49.6.41 Contrast And Brightness Control Register

Name: ISC_CBC_CTRL

Address: 0xF00083B4

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	ENABLE

- **ENABLE: Contrast and Brightness Control Enable**

0: Contrast and brightness control is disabled.

1: Contrast and brightness control is enabled.

49.6.42 Contrast And Brightness Configuration Register

Name: ISC_CBC_CFG

Address: 0xF00083B8

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	CCIRMODE		CCIR

- **CCIR: CCIR656 Stream Enable**

0: Raw mode

1: CCIR656 stream

- **CCIRMODE: CCIR656 Byte Ordering**

Value	Name	Description
0	CBY	Byte ordering Cb0, Y0, Cr0, Y1
1	CRY	Byte ordering Cr0, Y0, Cb0, Y1
2	YCB	Byte ordering Y0, Cb0, Y1, Cr0
3	YCR	Byte ordering Y0, Cr0, Y1, Cb0

49.6.43 Contrast And Brightness, Brightness Register

Name: ISC_CBC_BRIGHT

Address: 0xF00083BC

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	BRIGHT		
7	6	5	4	3	2	1	0
BRIGHT							

- **BRIGHT: Brightness Control (signed 11 bits 1:10:0)**

49.6.44 Contrast And Brightness, Contrast Register

Name: ISC_CBC_CONTRAST

Address: 0xF00083C0

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	–	–	CONTRAST			

7	6	5	4	3	2	1	0
CONTRAST							

- **CONTRAST: Contrast (signed 12 bits 1:3:8)**

49.6.45 Subsampling 4:4:4 to 4:2:2 Control Register

Name: ISC_SUB422_CTRL

Address: 0xF00083C4

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	ENABLE

- **ENABLE: 4:4:4 to 4:2:2 Chrominance Horizontal Subsampling Filter Enable**

0: Subsampler is disabled.

1: Subsampler is enabled.

49.6.46 Subsampling 4:4:4 to 4:2:2 Configuration Register

Name: ISC_SUB422_CFG

Address: 0xF00083C8

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	FILTER		–	CCIRMODE		CCIR

- **CCIR: CCIR656 Input Stream**

0: Raw mode

1: CCIR mode

- **CCIRMODE: CCIR656 Byte Ordering**

Value	Name	Description
0	CBY	Byte ordering Cb0, Y0, Cr0, Y1
1	CRY	Byte ordering Cr0, Y0, Cb0, Y1
2	YCB	Byte ordering Y0, Cb0, Y1, Cr0
3	YCR	Byte ordering Y0, Cr0, Y1, Cb0

- **FILTER: Low Pass Filter Selection**

Value	Name	Description
0	FILT0CO	Cosited, {1}
1	FILT1CE	Centered {1, 1}
2	FILT2CO	Cosited {1,2,1}
3	FILT3CE	Centered {1, 3, 3, 1}

49.6.47 Subsampling 4:2:2 to 4:2:0 Control Register

Name: ISC_SUB420_CTRL

Address: 0xF00083CC

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	FILTER	–	–	–	ENABLE

- **ENABLE: 4:2:2 to 4:2:0 Vertical Subsampling Filter Enable (Center Aligned)**

0: Subsampler disabled

1: Subsampler enabled

- **FILTER: Interlaced or Progressive Chrominance Filter**

0: Progressive filter {0.5, 0.5}

1: Field-dependent filter, top field filter is {0.75, 0.25}, bottom field filter is {0.25, 0.75}

49.6.48 Rounding, Limiting and Packing Configuration Register

Name: ISC_RLP_CFG

Address: 0xF00083D0

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
ALPHA							
7	6	5	4	3	2	1	0
–	–	–	–	MODE			

- **MODE: Rounding, Limiting and Packing Mode**

Value	Name	Description
0	DAT8	8-bit data
1	DAT9	9-bit data
2	DAT10	10-bit data
3	DAT11	11-bit data
4	DAT12	12-bit data
5	DATY8	8-bit luminance only
6	DATY10	10-bit luminance only
7	ARGB444	12-bit RGB+4-bit Alpha (MSB)
8	ARGB555	15-bit RGB+1-bit Alpha (MSB)
9	RGB565	16-bit RGB
10	ARGB32	24-bits RGB mode+8-bit Alpha
11	YYCC	YCbCr mode (full range, [0–255])
12	YYCC_LIMITED	YCbCr mode (limited range)

- **ALPHA: Alpha Value for Alpha-enabled RGB Mode**

49.6.49 Histogram Control Register

Name: ISC_HIS_CTRL

Address: 0xF00083D4

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	ENABLE

- **ENABLE: Histogram Sub Module Enable**

0: Histogram disabled.

1: Histogram enabled.

49.6.50 Histogram Configuration Register

Name: ISC_HIS_CFG

Address: 0xF00083D8

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	RAR
7	6	5	4	3	2	1	0
–	–	BAYSEL		–	MODE		

- **MODE: Histogram Operating Mode**

Value	Name	Description
0	Gr	Gr sampling
1	R	R sampling
2	Gb	Gb sampling
3	B	B sampling
4	Y	Luminance-only mode
5	RAW	Raw sampling
6	YCCIR656	Luminance only with CCIR656 10-bit or 8-bit mode

- **BAYSEL: Bayer Color Component Selection**

Value	Name	Description
0	GRGR	Starting row configuration is G R G R (red row)
1	RGRG	Starting row configuration is R G R G (red row)
2	GBGB	Starting row configuration is G B G B (blue row
3	BGBG	Starting row configuration is B G B G (blue row)

- **RAR: Histogram Reset After Read**

0: Reset after read mode is disabled

1: Reset after read mode is enabled

49.6.51 DMA Configuration Register

Name: ISC_DCFG

Address: 0xF00083E0

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	–	–	–	–	CMBSIZE	

7	6	5	4	3	2	1	0
–	–	YMBSIZE		–	IMODE		

- **IMODE: DMA Input Mode Selection**

Value	Name	Description
0	PACKED8	8 bits, single channel packed
1	PACKED16	16 bits, single channel packed
2	PACKED32	32 bits, single channel packed
3	YC422SP	32 bits, dual channel
4	YC422P	32 bits, triple channel
5	YC420SP	32 bits, dual channel
6	YC420P	32 bits, triple channel

- **YMBSIZE: DMA Memory Burst Size Y channel**

Value	Name	Description
0	SINGLE	DMA single access
1	BEATS4	4-beat burst access
2	BEATS8	8-beat burst access
3	BEATS16	16-beat burst access

- **CMBSIZE: DMA Memory Burst Size C channel**

Value	Name	Description
0	SINGLE	DMA single access
1	BEATS4	4-beat burst access
2	BEATS8	8-beat burst access
3	BEATS16	16-beat burst access

49.6.52 DMA Control Register

Name: ISC_DCTRL

Address: 0xF00083E4

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
DONE	FIELD	WB	IE	–	DVIEW		DE

- **DE: Descriptor Enable**

0: Descriptor disabled

1: Descriptor enabled

- **DVIEW: Descriptor View**

Value	Name	Description
0	PACKED	Address {0} Stride {0} are updated
1	SEMIPLANAR	Address {0,1} Stride {0,1} are updated
2	PLANAR	Address {0,1,2} Stride {0,1,2} are updated

- **IE: Interrupt Enable**

0: DMA Done interrupt is generated.

1: DMA Done interrupt is not set.

- **WB: Write Back Operation Enable**

0: Write Back operation is skipped.

1: Write Back operation is performed.

- **FIELD: Value of Captured Frame Field Signal**[1][2]

0: Field value is 0.

1: Field value is 1.

- **DONE: Descriptor Processing Status**[2]

0: Descriptor not processed yet

1: Descriptor processed

Notes: 1. Only relevant for interlaced content.

2. Appears only in descriptor located in memory. Can be used only if WB (Write Back) is set.

49.6.53 DMA Descriptor Address Register

Name: ISC_DNDA

Address: 0xF00083E8

Access: Read/Write

31	30	29	28	27	26	25	24
NDA							

23	22	21	20	19	18	17	16
NDA							

15	14	13	12	11	10	9	8
NDA							

7	6	5	4	3	2	1	0
NDA						–	–

- **NDA: Next Descriptor Address Register**

49.6.54 DMA Address 0 Register

Name: ISC_DAD0

Address: 0xF00083EC

Access: Read/Write

31	30	29	28	27	26	25	24
AD0							

23	22	21	20	19	18	17	16
AD0							

15	14	13	12	11	10	9	8
AD0							

7	6	5	4	3	2	1	0
AD0							

- **AD0: Channel 0 Address**

49.6.55 DMA Stride 0 Register

Name: ISC_DST0

Address: 0xF00083F0

Access: Read/Write

31	30	29	28	27	26	25	24
ST0							

23	22	21	20	19	18	17	16
ST0							

15	14	13	12	11	10	9	8
ST0							

7	6	5	4	3	2	1	0
ST0							

- **ST0: Channel 0 Stride**

49.6.56 DMA Address 1 Register

Name: ISC_DAD1

Address: 0xF00083F4

Access: Read/Write

31	30	29	28	27	26	25	24
AD1							

23	22	21	20	19	18	17	16
AD1							

15	14	13	12	11	10	9	8
AD1							

7	6	5	4	3	2	1	0
AD1							

- **AD1: Channel 1 Address**

49.6.57 DMA Stride 1 Register

Name: ISC_DST1

Address: 0xF00083F8

Access: Read/Write

31	30	29	28	27	26	25	24
ST1							

23	22	21	20	19	18	17	16
ST1							

15	14	13	12	11	10	9	8
ST1							

7	6	5	4	3	2	1	0
ST1							

- **ST1: Channel 1 Stride**

49.6.58 DMA Address 2 Register

Name: ISC_DAD2

Address: 0xF00083FC

Access: Read/Write

31	30	29	28	27	26	25	24
AD2							

23	22	21	20	19	18	17	16
AD2							

15	14	13	12	11	10	9	8
AD2							

7	6	5	4	3	2	1	0
AD2							

- **AD2: Channel 2 Address**

49.6.59 DMA Stride 2 Register

Name: ISC_DST2

Address: 0xF0008400

Access: Read/Write

31	30	29	28	27	26	25	24
ST2							

23	22	21	20	19	18	17	16
ST2							

15	14	13	12	11	10	9	8
ST2							

7	6	5	4	3	2	1	0
ST2							

- **ST2: Channel 2 Stride**

49.6.60 Histogram Entry

Name: ISC_HIS_ENTRYx [x=0..511]

Address: 0xF0008410

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	COUNT			

15	14	13	12	11	10	9	8
COUNT							

7	6	5	4	3	2	1	0
COUNT							

- **COUNT: Entry Counter**

50. Controller Area Network (MCAN)

50.1 Description

The Controller Area Network (MCAN) performs communication according to ISO11898-1 (Bosch CAN specification 2.0 part A,B) and to Bosch CAN FD specification V1.0. Additional transceiver hardware is required for connection to the physical layer.

All functions concerning the handling of messages are implemented by the Rx Handler and the Tx Handler. The Rx Handler manages message acceptance filtering, the transfer of received messages from the CAN core to the Message RAM, as well as providing receive message status information. The Tx Handler is responsible for the transfer of transmit messages from the Message RAM to the CAN core, as well as providing transmit status information.

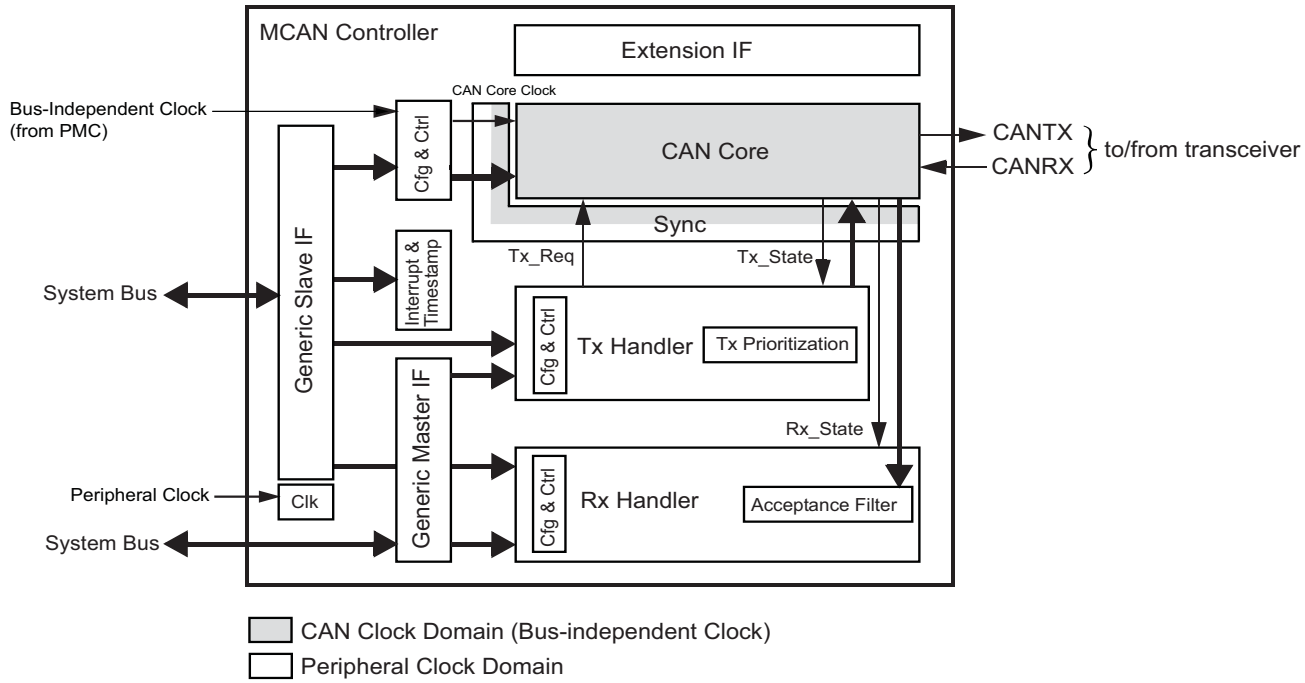
Acceptance filtering is implemented by a combination of up to 128 filter elements, where each element can be configured as a range, as a bit mask, or as a dedicated ID filter.

50.2 Embedded Characteristics

- Compliant with CAN Protocol Version 2.0 Part A, B and ISO 11898-1
- CAN FD with up to 64 Data Bytes Supported
- CAN Error Logging
- AUTOSAR Optimized
- SAE J1939 Optimized
- Improved Acceptance Filtering
- Two Configurable Receive FIFOs
- Separate Signalling on Reception of High Priority Messages
- Up to 64 Dedicated Receive Buffers
- Up to 32 Dedicated Transmit Buffers
- Configurable Transmit FIFO
- Configurable Transmit Queue
- Configurable Transmit Event FIFO
- Direct Message RAM Access for Processor
- Multiple MCANs May Share the Same Message RAM
- Programmable Loop-back Test Mode
- Maskable Module Interrupts
- Support for Asynchronous CAN and System Bus Clocks
- Power-down Support
- Debug on CAN Support

50.3 Block Diagram

Figure 50-1. MCAN Block Diagram



Note: Refer to section “Power Management Controller (PMC)” for details about the bus-independent clock (GCLK).

50.4 Product Dependencies

50.4.1 I/O Lines

The pins used to interface to the compliant external devices can be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the CAN pins to their peripheral functions.

Table 50-1. I/O Lines

Instance	Signal	I/O Line	Peripheral
MCAN0	CANRX0	PC2	C
MCAN0	CANRX0	PC11	E
MCAN0	CANTX0	PC1	C
MCAN0	CANTX0	PC10	E
MCAN1	CANRX1	PC27	D
MCAN1	CANTX1	PC26	D

50.4.2 Power Management

The MCAN can be clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the MCAN clock.

In order to achieve a stable function of the MCAN, the system bus clock must always be faster than or equal to the CAN clock.

It is recommended to use the CAN clock at frequencies of 20, 40 or 80 MHz. To achieve these frequencies, PMC GCLK must select the UPLLCK (480 MHz) as source clock and divide by 24,12, or 6. GCLK allows the system bus and processor clock to be modified without affecting the bit rate communication.

50.4.3 Interrupt Sources

The two MCAN interrupt lines (MCAN_INT0, MCAN_INT1) are connected on internal sources of the Interrupt Controller.

Using the MCAN interrupts requires the Interrupt Controller to be programmed first.

Interrupt sources can be routed either to MCAN_INT0 or to MCAN_INT1. By default, all interrupt sources are routed to interrupt line MCAN_INT0/1. By programming MCAN_ILE.EINT0 and MCAN_ILE.EINT1, the interrupt sources can be enabled or disabled separately.

Table 50-2. Peripheral IDs

Instance	ID
MCAN0	56
MCAN1	57

50.4.4 Address Configuration

The LSBs [bits 15:2] for each section of the CAN Message RAM are configured in the respective buffer configuration registers.

The MSBs [bits 31:16] of the CAN Message RAM for CAN0 and CAN1 are configured in 0x00200000.

50.5 Functional Description

50.5.1 Operating Modes

50.5.1.1 Software Initialization

Software initialization is started by setting bit MCAN_CCCR.INIT, either by software or by a hardware reset, when an uncorrected bit error was detected in the Message RAM, or by going Bus_Off. While MCAN_CCCR.INIT is set, message transfer from and to the CAN bus is stopped and the status of the CAN bus output CANTX is recessive (HIGH). The counters of the Error Management Logic EML are unchanged. Setting MCAN_CCCR.INIT does not change any configuration register. Resetting MCAN_CCCR.INIT finishes the software initialization. Afterwards the Bit Stream Processor BSP synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive recessive bits (≡ Bus_Idle) before it can take part in bus activities and start the message transfer.

Access to the MCAN configuration registers is only enabled when both bits MCAN_CCCR.INIT and MCAN_CCCR.CCE are set (protected write).

MCAN_CCCR.CCE can only be configured when MCAN_CCCR.INIT = ‘1’. MCAN_CCCR.CCE is automatically cleared when MCAN_CCCR.INIT = ‘0’.

The following registers are cleared when MCAN_CCCR.CCE = ‘1’:

- High Priority Message Status (MCAN_HPMS)
- Receive FIFO 0 Status (MCAN_RXF0S)
- Receive FIFO 1 Status (MCAN_RXF1S)
- Transmit FIFO/Queue Status (MCAN_TXFQS)
- Transmit Buffer Request Pending (MCAN_TXBRP)
- Transmit Buffer Transmission Occurred (MCAN_TXBTO)
- Transmit Buffer Cancellation Finished (MCAN_TXBCF)
- Transmit Event FIFO Status (MCAN_TXEFS)

The Timeout Counter value MCAN_TOCV.TOC is loaded with the value configured by MCAN_TOCC.TOP when MCAN_CCCR.CCE = ‘1’.

In addition, the state machines of the Tx Handler and Rx Handler are held in idle state while MCAN_CCCR.CCE = ‘1’.

The following registers are only writeable while MCAN_CCCR.CCE = ‘0’

- Transmit Buffer Add Request (MCAN_TXBAR)
- Transmit Buffer Cancellation Request (MCAN_TXBCR)

MCAN_CCCR.TEST and MCAN_CCCR.MON can only be set when MCAN_CCCR.INIT = ‘1’ and MCAN_CCCR.CCE = ‘1’. Both bits may be cleared at any time. MCAN_CCCR.DAR can only be configured when MCAN_CCCR.INIT = ‘1’ and MCAN_CCCR.CCE = ‘1’.

50.5.1.2 Normal Operation

Once the MCAN is initialized and MCAN_CCCR.INIT is cleared, the MCAN synchronizes itself to the CAN bus and is ready for communication.

After passing the acceptance filtering, received messages including Message ID and DLC are stored into a dedicated Rx Buffer or into Rx FIFO 0 or Rx FIFO 1.

For messages to be transmitted, dedicated Tx Buffers and/or a Tx FIFO or a Tx Queue can be initialized or updated. Automated transmission on reception of remote frames is not implemented.

50.5.1.3 CAN FD Operation

There are two variants in the CAN FD frame format, first the CAN FD frame without bit rate switching where the data field of a CAN frame may be longer than 8 bytes. The second variant is the CAN FD frame where control field, data field, and CRC field of a CAN frame are transmitted with a higher bit rate than the beginning and the end of the frame.

The previously reserved bit in CAN frames with 11-bit identifiers and the first previously reserved bit in CAN frames with 29-bit identifiers will now be decoded as FDF bit. FDF = recessive signifies a CAN FD frame, FDF = dominant signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF, res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive. The coding of res = recessive is reserved for future expansion of the protocol. In case the MCAN receives a frame with FDF = recessive and res = recessive, it will signal a Protocol Exception Event by setting bit MCAN_PSR.PXE. When Protocol Exception Handling is enabled (MCAN_CCCR.PXHD = 0), this causes the operation state to change from Receiver (MCAN_PSR.ACT = 2) to Integrating (MCAN_PSR.ACT = 00) at the next sample point. In case Protocol Exception Handling is disabled (MCAN_CCCR.PXHD = 1), the MCAN will treat a recessive res bit as an form error and will respond with an error frame.

CAN FD operation is enabled by programming CCCR.FDOE. In case CCCR.FDOE = ‘1’, transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured via bit FDF in the respective Tx Buffer element. With CCCR.FDOE = ‘0’, received frames are interpreted as Classic CAN frames, which leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if bit FDF of a Tx Buffer element is set. CCCR.FDOE and CCCR.BRSE can only be changed while CCCR.INIT and CCCR.CCE are both set.

With MCAN_CCCR.FDOE = 0, the setting of bits FDF and BRS is ignored and frames are transmitted in Classic CAN format. With MCAN_CCCR.FDOE = 1 and MCAN_CCCR.BRSE = 0, only bit FDF of a Tx Buffer element is evaluated. With MCAN_CCCR.FDOE = 1 and MCAN_CCCR.BRSE = 1, transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with bits FDF and BRS set are transmitted in CAN FD format with bit rate switching.

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significant higher than in the CAN FD arbitration phase. In this case disable the CAN FD bit rate switching option for transmissions.
- During system startup all nodes are transmitting according to ISO11898-1 until it is verified that they are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wakeup messages in CAN Partial Networking have to be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD-capable. Non-CAN FD nodes are held in Silent mode until programming has completed. Then all nodes revert to Classic CAN communication.

In the CAN FD format, the coding of the DLC differs from the standard CAN format. The DLC codes 0 to 8 have the same coding as in standard CAN, the codes 9 to 15, which in standard CAN all code a data field of 8 bytes, are coded according to Table 50-3 below.

Table 50-3. Coding of DLC in CAN FD

DLC	9	10	11	12	13	14	15
Number of Data Bytes	12	16	20	24	32	48	64

In CAN FD frames, the bit timing will be switched inside the frame, after the BRS (Bit Rate Switch) bit, if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal CAN bit timing is used as defined by the Nominal Bit Timing and Prescaler register (MCAN_NBTP). In the following CAN FD data phase, the fast CAN bit timing is used as defined by the Data Bit Timing and Prescaler register (MCAN_DBTP). The bit timing reverts back from the fast timing at the CRC delimiter or when an error is detected, whichever occurs first.

The maximum configurable bit rate in the CAN FD data phase depends on the CAN core clock frequency. Example: with a CAN clock frequency of 20 MHz and the shortest configurable bit time of 4 t_q, the bit rate in the data phase is 5 Mbit/s.

In both data frame formats, CAN FD and CAN FD with bit rate switching, the value of the bit ESI (Error Status Indicator) is determined by the transmitter's error state at the start of the transmission. If the transmitter is error passive, ESI is transmitted recessive, else it is transmitted dominant.

50.5.1.4 Transmitter Delay Compensation

During the data phase of a CAN FD transmission only one node is transmitting, all others are receivers. The length of the bus line has no impact. When transmitting via pin CANTX the protocol controller receives the transmitted data from its local CAN transceiver via pin CANRX. The received data is delayed by the transmitter delay. In case this delay is greater than NTSEG1 (time segment before sample point), a bit error is detected. In order to enable a data phase bit time that is even shorter than the transmitter delay, the delay compensation is introduced. Without transmitter delay compensation, the bit rate in the data phase of a CAN FD frame is limited by the transmitter delay.

Description

The MCAN protocol unit has implemented a delay compensation mechanism to compensate the transmitter delay, thereby enabling transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver.

To check for bit errors during the data phase, the delayed transmit data is compared against the received data at the secondary sample point. If a bit error is detected, the transmitter will react to this bit error at the next following regular sample point. During arbitration phase the delay compensation is always disabled.

The transmitter delay compensation enables configurations where the data bit time is shorter than the transmitter delay, it is described in detail in the new ISO11898-1. It is enabled by setting bit MCAN_DBTP.TDC.

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the MCAN's transmit output CANTX through the transceiver to the receive input CANRX plus the transmitter delay compensation offset as configured by MCAN_TDCR.TDCO. The transmitter delay compensation offset is used to adjust the position of the SSP inside the received bit (e.g. half of the bit time in the data phase). The position of the secondary sample point is rounded down to the next integer number of CAN core clock periods.

MCAN_PSR.TDCV shows the actual transmitter delay compensation value. MCAN_PSR.TDCV is cleared when MCAN_CCCR.INIT is set and is updated at each transmission of an FD frame while MCAN_DBTP.TDC is set.

The following boundary conditions have to be considered for the transmitter delay compensation implemented in the MCAN:

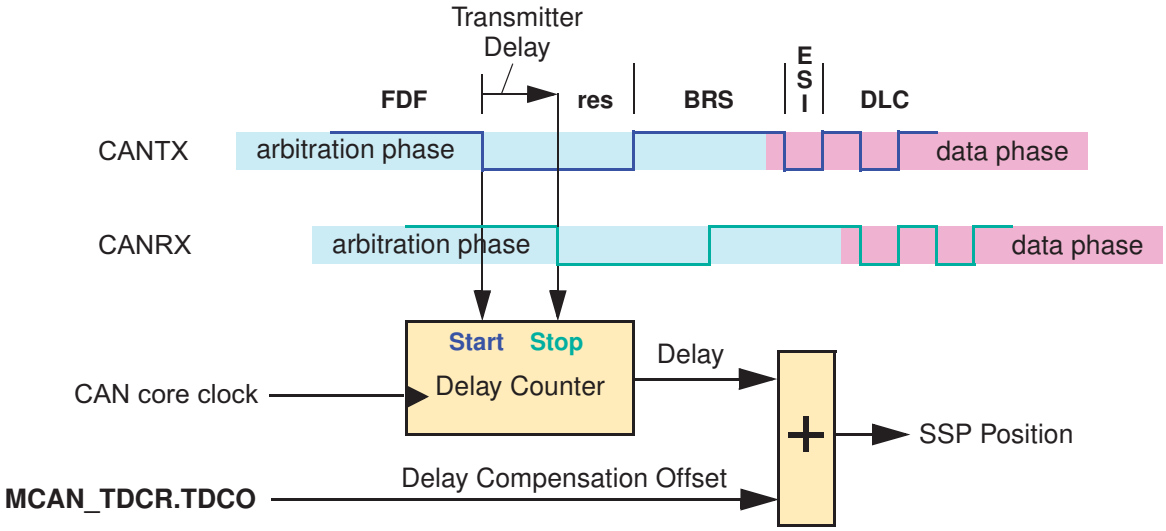
- The sum of the measured delay from CANTX to CANRX and the configured transceiver delay compensation offset MCAN_TDCR.TDCO has to be less than 6 bit times in the data phase.
- The sum of the measured delay from CANTX to CANRX and the configured transceiver delay compensation offset MCAN_TDCR.TDCO has to be less or equal 127 CAN core clock periods. In case this sum exceeds 127 CAN core clock periods, the maximum value of 127 CAN core clock periods is used for transceiver delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of receive bits at the SSPs.

Transmitter Delay Measurement

If transmitter delay compensation is enabled by programming MCAN_DBTP.TDC = '1', the measurement is started within each transmitted CAN FD frame at the falling edge of bit FDF to bit res. The measurement is stopped when this edge is seen at the receive input CANRX of the transmitter.

The resolution of this measurement is one mtq.

Figure 50-2. Transmitter Delay Measurement



To avoid that a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in a to early SSP position, the use of a transmitter delay compensation filter window can be enabled by programming MCAN_TDCR.TDCF.

This defines a minimum value for the SSP position. Dominant edges on CANRX, that would result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least MCAN_TDCR.TDCF AND CANRX is low.

50.5.1.5 Restricted Operation Mode

In Restricted Operation mode, the node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The error counters are not incremented. The processor can set the MCAN into Restricted Operation mode by setting bit MCAN_CCCR.ASM. The bit can only be set by the processor when both MCAN_CCCR.CCE and MCAN_CCCR.INIT are set to '1'. The bit can be reset by the processor at any time.

Restricted Operation mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation mode, the processor has to reset MCAN_CCCR.ASM.

The Restricted Operation mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation mode after it has received a valid frame.

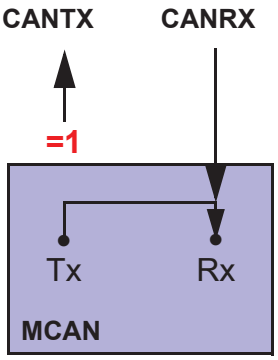
Note: The Restricted Operation Mode must not be combined with the Loop Back mode (internal or external).

50.5.1.6 Bus Monitoring Mode

The MCAN is set in Bus Monitoring mode by setting MCAN_CCCR.MON. In Bus Monitoring mode (see ISO11898-1, 10.12 Bus monitoring), the MCAN is able to receive valid data frames and valid remote frames, but cannot start a transmission. In this mode, it sends only recessive bits on the CAN bus. If the MCAN is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the MCAN monitors this dominant bit, although the CAN bus may remain in recessive state. In Bus Monitoring mode, the Tx Buffer Request Pending register (MCAN_TXBRP) is held in reset state.

The Bus Monitoring mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. Figure 50-4 shows the connection of signals CANTX and CANRX to the MCAN in Bus Monitoring mode.

Figure 50-3. Pin Control in Bus Monitoring Mode



Bus Monitoring Mode

50.5.1.7 Disabled Automatic Retransmission

According to the CAN Specification (see ISO11898-1, 6.3.3 Recovery Management), the MCAN provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled. To support time-triggered communication as described in ISO 11898-1, chapter 9.2, the automatic retransmission may be disabled via MCAN_CCCR.DAR.

Frame Transmission in DAR Mode

In DAR mode, all transmissions are automatically cancelled after they start on the CAN bus. A Tx Buffer's Tx Request Pending bit TXBRP.TRPx is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

- Successful transmission:
  Corresponding Tx Buffer Transmission Occurred bit MCAN_TXBTO.TOx set
  Corresponding Tx Buffer Cancellation Finished bit MCAN_TXBCF.CFx not set
- Successful transmission in spite of cancellation:
  Corresponding Tx Buffer Transmission Occurred bit MCAN_TXBTO.TOx set
  Corresponding Tx Buffer Cancellation Finished bit MCAN_TXBCF.CFx set
- Arbitration lost or frame transmission disturbed:
  Corresponding Tx Buffer Transmission Occurred bit MCAN_TXBTO.TOx not set
  Corresponding Tx Buffer Cancellation Finished bit MCAN_TXBCF.CFx set

In case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = “10” (transmission in spite of cancellation).

50.5.1.8 Power-down (Sleep Mode)

The MCAN can be set into Power-down mode via bit MCAN_CCCR.CSR.

When all pending transmission requests have completed, the MCAN waits until bus idle state is detected. Then the MCAN sets MCAN_CCCR.INIT to prevent any further CAN transfers. Now the MCAN acknowledges that it is ready for power down by setting to one the bit MCAN_CCCR.CSA. In this state, before the clocks are switched off, further register accesses can be made. A write access to MCAN_CCCR.INIT will have no effect. Now the bus clock (peripheral clock) and the CAN core clock may be switched off.

To leave Power-down mode, the application has to turn on the MCAN clocks before clearing CC Control Register flag MCAN_CCCR.CSR. The MCAN will acknowledge this by clearing MCAN_CCCR.CSA. The application can then restart CAN communication by clearing the bit CCCR.INIT.

50.5.1.9 Test Modes

To enable write access to the MCAN Test register (MCAN_TEST) (see Section 50.6.5), bit MCAN_CCCR.TEST must be set. This allows the configuration of the test modes and test functions.

Four output functions are available for the CAN transmit pin CANTX by programming MCAN_TEST.TX. Additionally to its default function – the serial data output – it can drive the CAN Sample Point signal to monitor the MCAN's bit timing and it can drive constant dominant or recessive values. The actual value at pin CANRX can be read from MCAN_TEST.RX. Both functions can be used to check the CAN bus' physical layer.

Due to the synchronization mechanism between CAN clock and system bus clock domain, there may be a delay of several system bus clock periods between writing to MCAN_TEST.TX until the new configuration is visible at output pin CANTX. This applies also when reading input pin CANRX via MCAN_TEST.RX.

Note: Test modes should be used for production tests or self-test only. The software control for pin CANTX interferes with all CAN protocol functions. It is not recommended to use test modes for application.

External Loop Back Mode

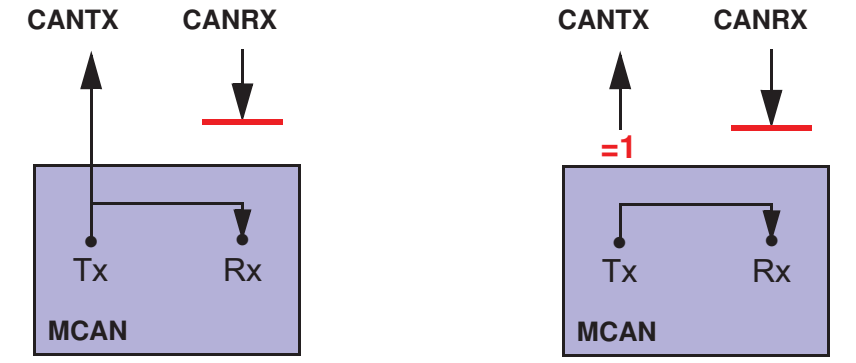
The MCAN can be set in External Loop Back mode by setting the bit MCAN_TEST.LBCK. In Loop Back mode, the MCAN treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into an Rx Buffer or an Rx FIFO. Figure 50-4 shows the connection of signals CANTX and CANRX to the MCAN in External Loop Back mode.

This mode is provided for hardware self-test. To be independent from external stimulation, the MCAN ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remote frame) in Loop Back mode. In this mode, the MCAN performs an internal feedback from its Tx output to its Rx input. The actual value of the CANRX input pin is disregarded by the MCAN. The transmitted messages can be monitored at the CANTX pin.

Internal Loop Back Mode

Internal Loop Back mode is entered by setting bits MCAN_TEST.LBCK and MCAN_CCCR.MON. This mode can be used for a "Hot Selftest", meaning the MCAN can be tested without affecting a running CAN system connected to the pins CANTX and CANRX. In this mode, pin CANRX is disconnected from the MCAN, and pin CANTX is held recessive. Figure 50-4 shows the connection of CANTX and CANRX to the MCAN when Internal Loop Back mode is enabled.

Figure 50-4. Pin Control in Loop Back Modes



50.5.2 Timestamp Generation

For timestamp generation the MCAN supplies a 16-bit wrap-around counter. A prescaler TSCC.TCP can be configured to clock the counter in multiples of CAN bit times (1...16). The counter is readable via MCAN_TSCV.TSC. A write access to the Timestamp Counter Value register (MCAN_TSCV) resets the counter to zero. When the timestamp counter wraps around, interrupt flag MCAN_IR.TSW is set.

On start of frame reception / transmission the counter value is captured and stored into the timestamp section of an Rx Buffer / Rx FIFO (RXTS[15:0]) or Tx Event FIFO (TXTS[15:0]) element.

By programming bit MCAN_TSCC.TSS an external 16-bit timestamp can be used.

50.5.3 Timeout Counter

To signal timeout conditions for Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO, the MCAN supplies a 16-bit Timeout Counter. It operates as down-counter and uses the same prescaler controlled by TSCC.TCP as the Timestamp Counter. The Timeout Counter is configured via the Timeout Counter Configuration register (MCAN_TOCC). The actual counter value can be read from MCAN_TOCV.TOC. The Timeout Counter can only be started while MCAN_CCCR.INIT = '0'. It is stopped when MCAN_CCCR.INIT = '1', e.g. when the MCAN enters Bus_Off state.

The operating mode is selected by MCAN_TOCC.TOS. When operating in Continuous mode, the counter starts when MCAN_CCCR.INIT is reset. A write to MCAN_TOCV presets the counter to the value configured by MCAN_TOCC.TOP and continues down-counting.

When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by MCAN_TOCC.TOP. Down-counting is started when the first FIFO element is stored. Writing to MCAN_TOCV has no effect.

When the counter reaches zero, interrupt flag MCAN_IR.TOO is set. In Continuous mode, the counter is immediately restarted at MCAN_TOCC.TOP.

Note: The clock signal for the Timeout Counter is derived from the CAN Core's sample point signal. Therefore the point in time where the Timeout Counter is decremented may vary due to the synchronization / re-synchronization mechanism of the CAN Core. If the bit rate switch feature in CAN FD is used, the timeout counter is clocked differently in arbitration and data field.

50.5.4 Rx Handling

The Rx Handler controls the acceptance filtering, the transfer of received messages to the Rx Buffers or to one of the two Rx FIFOs, as well as the Rx FIFO's Put and Get Indices.

50.5.4.1 Acceptance Filtering

The MCAN offers the possibility to configure two sets of acceptance filters, one for standard identifiers and one for extended identifiers. These filters can be assigned to an Rx Buffer or to Rx FIFO 0,1. For acceptance filtering each list of filters is executed from element #0 until the first matching element. Acceptance filtering stops at the first matching element. The following filter elements are not evaluated for this message.

The main features are:

- Each filter element can be configured as
  - range filter (from - to)
  - filter for one or two dedicated IDs
  - classic bit mask filter
- Each filter element is configurable for acceptance or rejection filtering
- Each filter element can be enabled / disabled individually
- Filters are checked sequentially, execution stops with the first matching filter element

Related configuration registers are:

- Global Filter Configuration (MCAN_GFC)
- Standard ID Filter Configuration (MCAN_SIDFC)
- Extended ID Filter Configuration (MCAN_XIDFC)
- Extended ID and Mask (MCAN_XIDAM)

Depending on the configuration of the filter element (SFEC/EFEC) a match triggers one of the following actions:

- Store received frame in FIFO 0 or FIFO 1
- Store received frame in Rx Buffer
- Store received frame in Rx Buffer and generate pulse at filter event pin
- Reject received frame
- Set High Priority Message interrupt flag (MCAN_IR.HPM)
- Set High Priority Message interrupt flag (MCAN_IR.HPM) and store received frame in FIFO 0 or FIFO 1

Acceptance filtering is started after the complete identifier has been received. After acceptance filtering has completed, and if a matching Rx Buffer or Rx FIFO has been found, the Message Handler starts writing the received message data in portions of 32 bit to the matching Rx Buffer or Rx FIFO. If the CAN protocol controller has detected an error condition (e.g. CRC error), this message is discarded with the following impact on the effected Rx Buffer or Rx FIFO:

- Rx Buffer
  New Data flag of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data. For error type, see MCAN_PSR.LEC and MCAN_PSR.DLEC.
- Rx FIFO
  Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data. For error type, see MCAN_PSR.LEC and MCAN_PSR.DLEC. In case the matching Rx FIFO is operated in Overwrite mode, the boundary conditions described in “Rx FIFO Overwrite Mode” have to be considered.

Note: When an accepted message is written to one of the two Rx FIFOs, or into an Rx Buffer, the unmodified received identifier is stored independent of the filter(s) used. The result of the acceptance filter process is strongly depending on the sequence of configured filter elements.

Range Filter

The filter matches for all received frames with Message IDs in the range defined by SF1ID/SF2ID resp. EF1ID/EF2ID.

There are two possibilities when range filtering is used together with extended frames:

EFT = “00”: The Message ID of received frames is ANDed with MCAN_XIDAM before the range filter is applied.

EFT = “11”: MCAN_XIDAM is not used for range filtering.

Filter for Specific IDs

A filter element can be configured to filter for one or two specific Message IDs. To filter for one specific Message ID, the filter element has to be configured with SF1ID = SF2ID resp. EF1ID = EF2ID.

Classic Bit Mask Filter

Classic bit mask filtering is intended to filter groups of Message IDs by masking single bits of a received Message ID. With classic bit mask filtering SF1ID/EF1ID is used as Message ID filter, while SF2ID/EF2ID is used as filter mask.

A zero bit at the filter mask will mask out the corresponding bit position of the configured ID filter, e.g. the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are one are relevant for acceptance filtering.

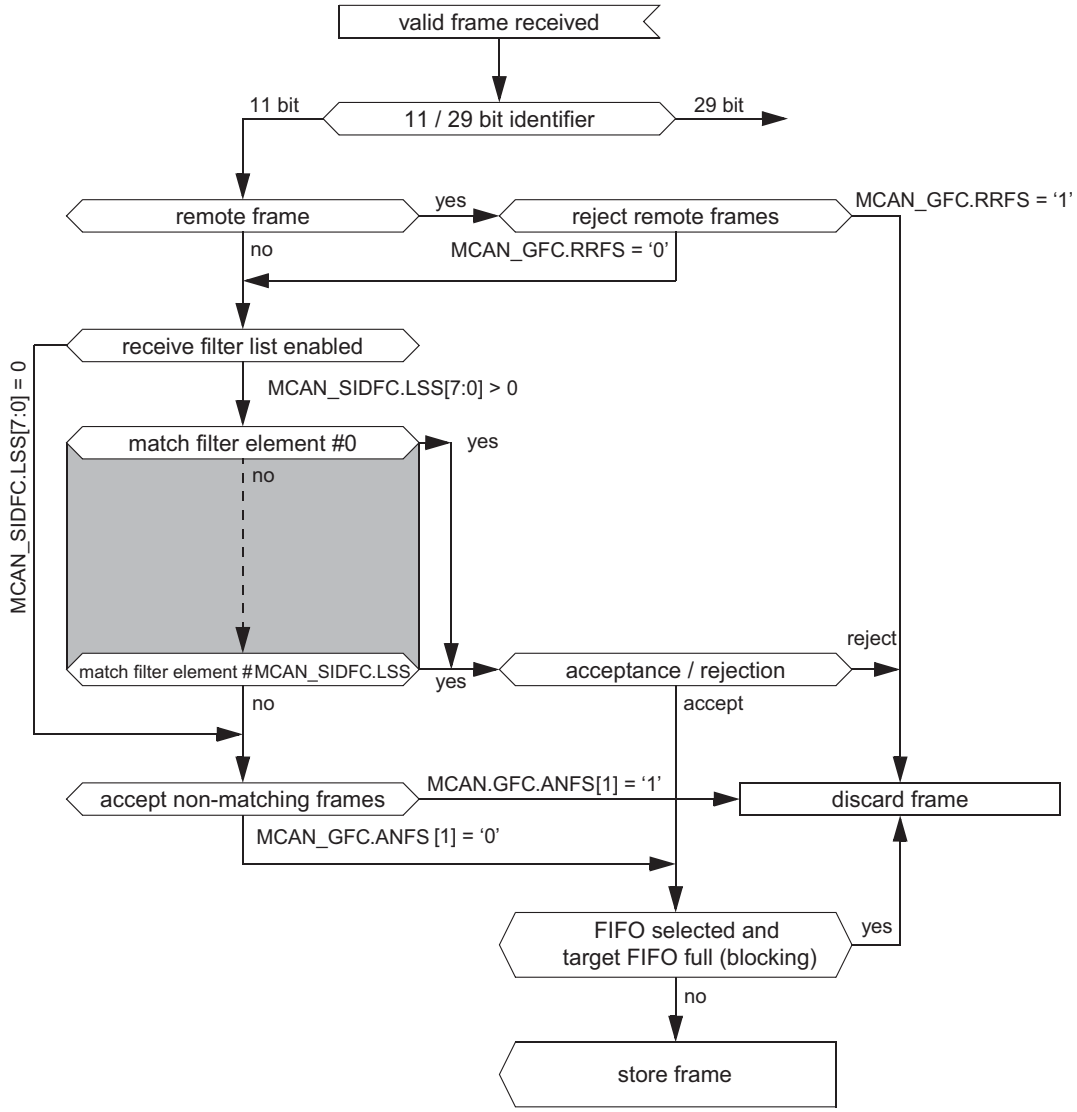
In case all mask bits are one, a match occurs only when the received Message ID and the Message ID filter are identical. If all mask bits are zero, all Message IDs match.

Standard Message ID Filtering

Figure 50-5 below shows the flow for standard Message ID (11-bit Identifier) filtering. The Standard Message ID Filter element is described in Section 50.5.7.5.

Controlled by MCAN_GFC and MCAN_SIDFC Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

Figure 50-5. Standard Message ID Filter Path



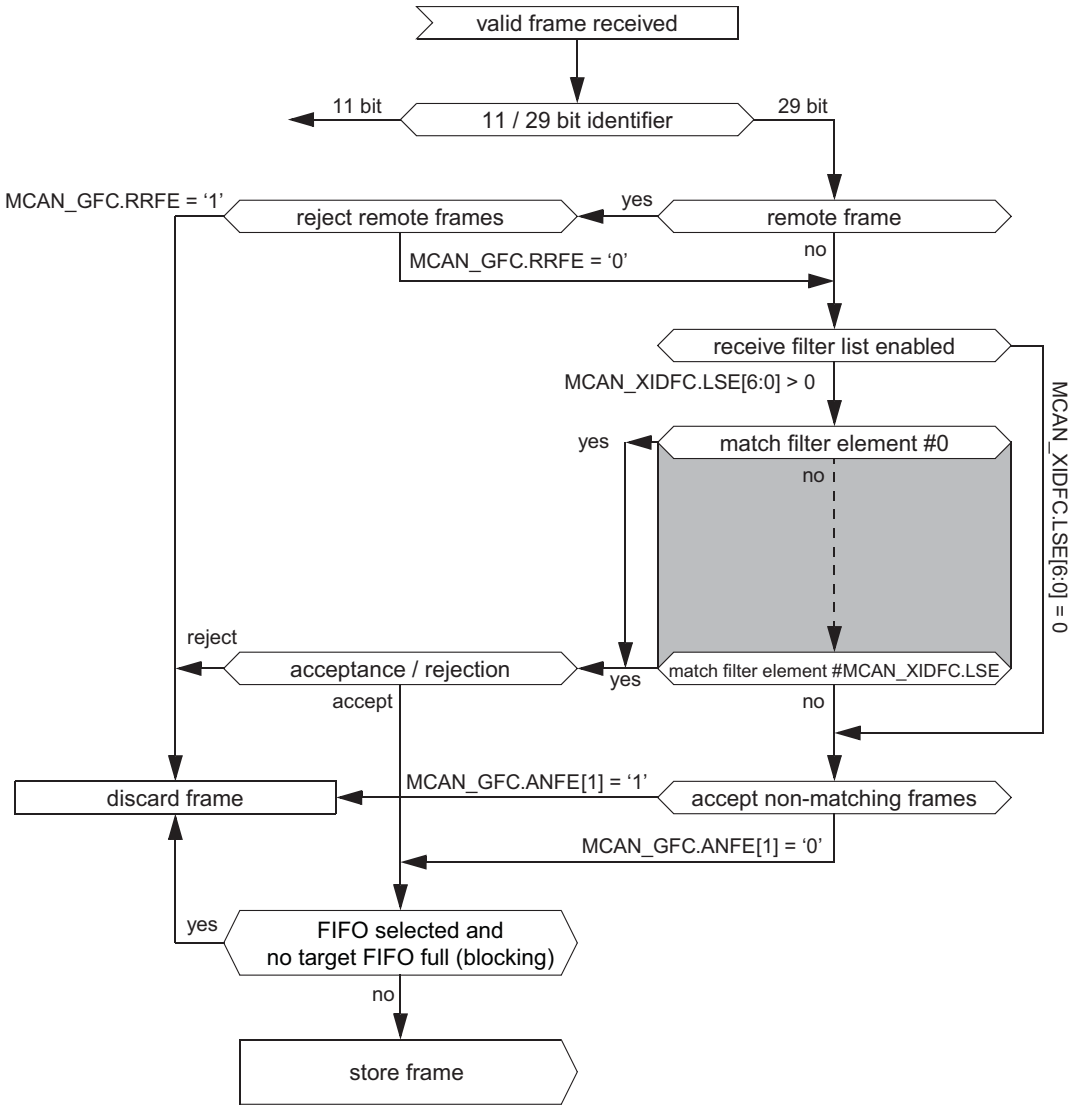
Extended Message ID Filtering

Figure 50-6 below shows the flow for extended Message ID (29-bit Identifier) filtering. The Extended Message ID Filter element is described in Section 50.5.7.6.

Controlled by MCAN_GFC and MCAN_XIDFC Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

MCAN_XIDAM is ANDed with the received identifier before the filter list is executed.

Figure 50-6. Extended Message ID Filter Path



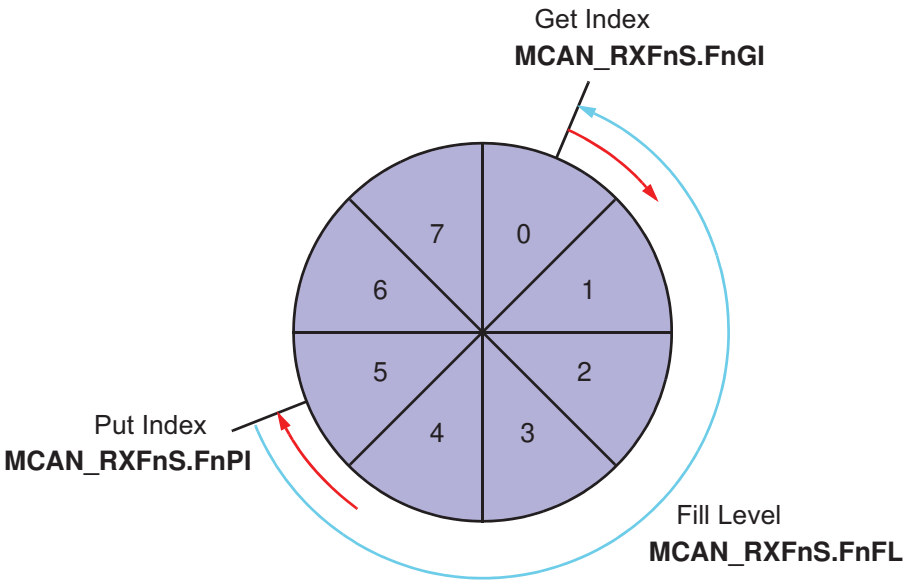
50.5.4.2 Rx FIFOs

Rx FIFO 0 and Rx FIFO 1 can be configured to hold up to 64 elements each. Configuration of the two Rx FIFOs is done via the Rx FIFO 0 Configuration register (MCAN_RXF0C) and the Rx FIFO 1 Configuration register (MCAN_RXF1C).

Received messages that passed acceptance filtering are transferred to the Rx FIFO as configured by the matching filter element. For a description of the filter mechanisms available for Rx FIFO 0 and Rx FIFO 1, see Section 50.5.4.1. The Rx FIFO element is described in Section 50.5.7.2.

To avoid an Rx FIFO overflow, the Rx FIFO watermark can be used. When the Rx FIFO fill level reaches the Rx FIFO watermark configured by MCAN_RXFnC.FnWM, interrupt flag MCAN_IR.RFnW is set. When the Rx FIFO Put Index reaches the Rx FIFO Get Index, an Rx FIFO Full condition is signalled by MCAN_RXFnS.FnF. In addition, the interrupt flag MCAN_IR.RFnF is set.

Figure 50-7. Rx FIFO Status



When reading from an Rx FIFO, Rx FIFO Get Index MCAN_RXFnS.FnGI × FIFO Element Size has to be added to the corresponding Rx FIFO start address MCAN_RXFnC.FnSA.

Table 50-4. Rx Buffer / FIFO Element Size

MCAN_RXESC.RBDS[2:0] MCAN_RXESC.FnDS[2:0]	Data Field [bytes]	FIFO Element Size [RAM words]
0	8	4
1	12	5
2	16	6
3	20	7
4	24	8
5	32	10
6	48	14
7	64	18

Rx FIFO Blocking Mode

The Rx FIFO Blocking mode is configured by MCAN_RXFnC.FnOM = ‘0’. This is the default operating mode for the Rx FIFOs.

When an Rx FIFO full condition is reached (MCAN_RXFnS.FnPI = MCAN_RXFnS.FnGI), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signalled by MCAN_RXFnS.FnF = ‘1’. In addition, the interrupt flag MCAN_IR.RFnF is set.

In case a message is received while the corresponding Rx FIFO is full, this message is discarded and the message lost condition is signalled by MCAN_RXFnS.RFnL = ‘1’. In addition, the interrupt flag MCAN_IR.RFnL is set.

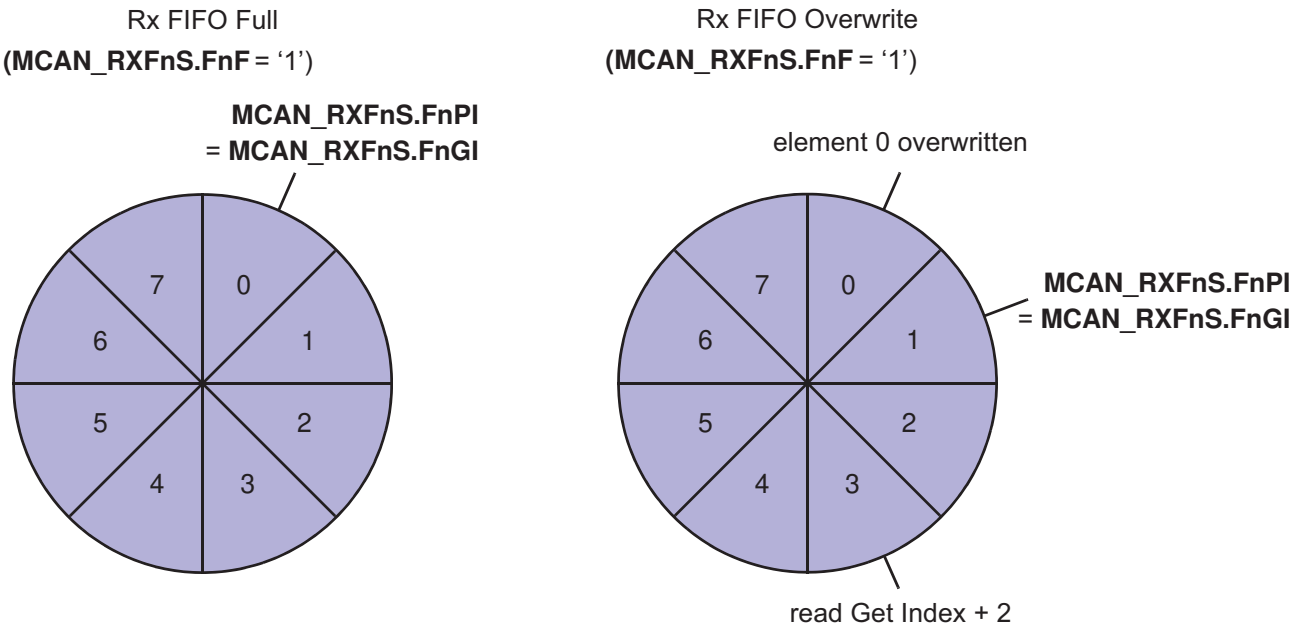
Rx FIFO Overwrite Mode

The Rx FIFO Overwrite mode is configured by MCAN_RXFnC.FnOM = ‘1’.

When an Rx FIFO full condition (MCAN_RXFnS.FnPI = MCAN_RXFnS.FnGI) is signalled by MCAN_RXFnS.FnF = ‘1’, the next message accepted for the FIFO will overwrite the oldest FIFO message. Put and get index are both incremented by one.

When an Rx FIFO is operated in Overwrite mode and an Rx FIFO full condition is signalled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might happen, that a received message is written to the Message RAM (put index) while the processor is reading from the Message RAM (get index). In this case inconsistent data may be read from the respective Rx FIFO element. Adding an offset to the get index when reading from the Rx FIFO avoids this problem. The offset depends on how fast the processor accesses the Rx FIFO. Figure 50-8 shows an offset of two with respect to the get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

Figure 50-8. Rx FIFO Overflow Handling



After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index MCAN_RXFnA.FnA. This increments the get index to that element number. In case the put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (MCAN_RXFnS.FnF = ‘0’).

50.5.4.3 Dedicated Rx Buffers

The MCAN supports up to 64 dedicated Rx Buffers. The start address of the dedicated Rx Buffer section is configured via MCAN_RXBC.RBSA.

For each Rx Buffer, a Standard or Extended Message ID Filter Element with SFEC / EFEC = 7 and SFID2 / EFID2[10:9] = 0 has to be configured (see Section 50.5.7.5 and Section 50.5.7.6).

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element. The format is the same as for an Rx FIFO element. In addition, the flag MCAN_IR.DRX (Message stored in dedicated Rx Buffer) in MCAN_IR is set.

Table 50-5. Example Filter Configuration for Rx Buffers

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID message 1	0	0
1	ID message 2	0	1
2	ID message 3	0	2

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in the New Data 1 register (MCAN_NDAT1) and New Data 2 register (MCAN_NDAT2) is set. As long as the New Data flag is set, the respective Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the processor by writing a '1' to the respective bit position.

While an Rx Buffer's New Data flag is set, a Message ID Filter Element referencing this specific Rx Buffer will not match, causing the acceptance filtering to continue. Following Message ID Filter Elements may cause the received message to be stored into another Rx Buffer, or into an Rx FIFO, or the message may be rejected, depending on filter configuration.

Rx Buffer Handling

- Reset interrupt flag IR.DRX
- Read New Data registers
- Read messages from Message RAM
- Reset New Data flags of processed messages

50.5.4.4 Debug on CAN Support

Debug messages are stored into Rx Buffers. For debug handling three consecutive Rx buffers (e.g. #61, #62, #63) have to be used for storage of debug messages A, B, and C. The format is the same as for an Rx Buffer or an Rx FIFO element (see Section 50.5.7.2 "Rx Buffer and FIFO Element").

Advantage: Fixed start address for the DMA transfers (relative to MCAN_RXBC.RBSA), no additional configuration required.

For filtering of debug messages Standard / Extended Filter Elements with SFEC / EFEC = '111' have to be set up. Messages matching these filter elements are stored into the Rx Buffers addressed by SFID2 / EFID2[5:0].

After message C has been stored, the DMA request output m_can_dma_req is activated and the three messages can be read from the Message RAM under DMA control. The RAM words holding the debug messages will not be changed by the MCAN while m_can_dma_req is activated. The behavior is similar to that of an Rx Buffer with its New Data flag set.

After the DMA has completed, the MCAN is prepared to receive the next set of debug messages.

Filtering for Debug Messages

Filtering for debug messages is done by configuring one Standard / Extended Message ID Filter Element for each of the three debug messages. To enable a filter element to filter for debug messages SFEC / EFEC has to be

programmed to “111”. In this case fields SFID1 / SFID2 and EFID1 / EFID2 have a different meaning (see Section 50.5.7.5 and Section 50.5.7.6). While SFID2 / EFID2[10:9] controls the debug message handling state machine, SFID2 / EFID2[5:0] controls the location for storage of a received debug message.

When a debug message is stored, neither the respective New Data flag nor MCAN_IR.DRX are set. The reception of debug messages can be monitored via RXF1S.DMS.

Table 50-6. Example Filter Configuration for Debug Messages

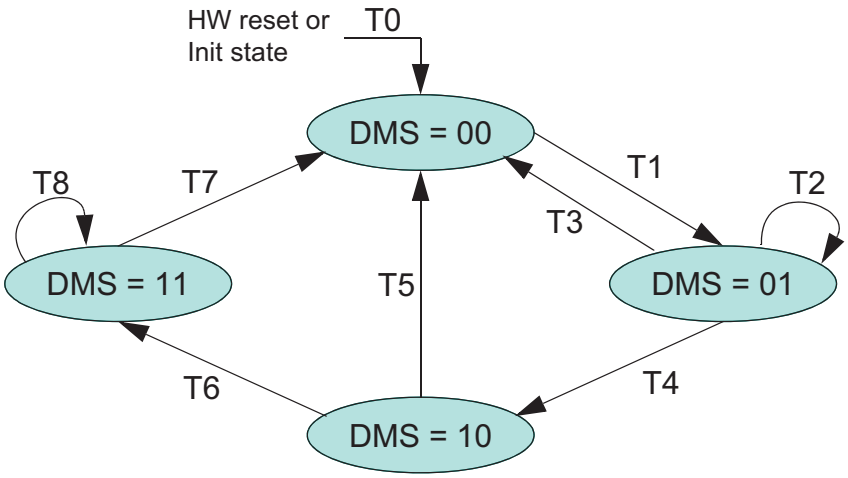
Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID debug message A	1	11 1101
1	ID debug message B	2	11 1110
2	ID debug message C	3	11 1111

Debug Message Handling

The debug message handling state machine ensures that debug messages are stored to three consecutive Rx Buffers in the correct order. If some messages are missing, the process is restarted. The DMA request is activated only when all three debug messages A, B, C have been received in the correct order.

The status of the debug message handling state machine is signalled via MCAN_RXF1S.DMS.

Figure 50-9. Debug Message Handling State Machine



T0: reset m_can_dma_req output, enable reception of debug messages A, B, and C

T1: reception of debug message A

T2: reception of debug message A

T3: reception of debug message C

T4: reception of debug message B

T5: reception of debug messages A, B

T6: reception of debug message C

T7: DMA transfer completed

T8: reception of debug message A,B,C (message rejected)

50.5.5 Tx Handling

The Tx Handler handles transmission requests for the dedicated Tx Buffers, the Tx FIFO, and the Tx Queue. It controls the transfer of transmit messages to the CAN Core, the Put and Get Indices, and the Tx Event FIFO. Up to 32 Tx Buffers can be set up for message transmission. The CAN mode for transmission (Classic CAN or CAN FD) can be configured separately for each Tx Buffer element. The Tx Buffer element is described in Section 50.5.7.3. Table 50-7 describes the possible configurations for frame transmission.

Table 50-7. Possible Configurations for Frame Transmission

MCAN_CCCR		Tx Buffer Element		
BRSE	**FDOE**	**FDF**	**BRS**	**Frame Transmission**
ignored	0	ignored	ignored	Classic CAN
0	1	0	ignored	Classic CAN
0	1	1	ignored	FD without bit rate switching
1	1	0	ignored	Classic CAN
1	1	1	0	FD without bit rate switching
1	1	1	1	FD with bit rate switching

Note: AUTOSAR requires at least three Tx Queue Buffers and support of transmit cancellation.

The Tx Handler starts a Tx scan to check for the highest priority pending Tx request (Tx Buffer with lowest Message ID) when MCAN_TXBRP is updated, or when a transmission has been started.

50.5.5.1 Transmit Pause

The transmit pause feature is intended for use in CAN systems where the CAN message identifiers are (permanently) specified to specific values and cannot easily be changed. These message identifiers may have a higher CAN arbitration priority than other defined messages, while in a specific application their relative arbitration priority should be inverse. This may lead to a case where one ECU sends a burst of CAN messages that cause another ECU's CAN messages to be delayed because that other messages have a lower CAN arbitration priority.

If e.g. CAN ECU-1 has the transmit pause feature enabled and is requested by its application software to transmit four messages, it will, after the first successful message transmission, wait for two CAN bit times of bus idle before it is allowed to start the next requested message. If there are other ECUs with pending messages, those messages are started in the idle time, they would not need to arbitrate with the next message of ECU-1. After having received a message, ECU-1 is allowed to start its next transmission as soon as the received message releases the CAN bus.

The transmit pause feature is controlled by bit MCAN_CCCR.TXP. If the bit is set, the MCAN will, each time it has successfully transmitted a message, pause for two CAN bit times before starting the next transmission. This enables other CAN nodes in the network to transmit messages even if their messages have lower prior identifiers. Default is transmit pause disabled (MCAN_CCCR.TXP = '0').

This feature looses up burst transmissions coming from a single node and it protects against "babbling idiot" scenarios where the application program erroneously requests too many transmissions.

50.5.5.2 Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the processor. Each dedicated Tx Buffer is configured with a specific Message ID. In case that multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first.

If the data section has been updated, a transmission is requested by an Add Request via MCAN_TXBAR.ARn. The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

A dedicated Tx Buffer allocates Element Size 32-bit words in the Message RAM (see Table 50-8). Therefore the start address of a dedicated Tx Buffer in the Message RAM is calculated by adding transmit buffer index (0…31) × Element Size to the Tx Buffer Start Address TXBC.TBSA.

Table 50-8. Tx Buffer / FIFO / Queue Element Size

TXESC.TBDS[2:0]	Data Field [bytes]	Element Size [RAM words]
0	8	4
1	12	5
2	16	6
3	20	7
4	24	8
5	32	10
6	48	14
7	64	18

50.5.5.3 Tx FIFO

Tx FIFO operation is configured by programming MCAN_TXBC.TFQM to ‘0’. Messages stored in the Tx FIFO are transmitted starting with the message referenced by the Get Index MCAN_TXFQS.TFGI. After each transmission the Get Index is incremented cyclically until the Tx FIFO is empty. The Tx FIFO enables transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO. The MCAN calculates the Tx FIFO Free Level MCAN_TXFQS.TFFL as difference between Get and Put Index. It indicates the number of available (free) Tx FIFO elements.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index MCAN_TXFQS.TFQPI. An Add Request increments the Put Index to the next free Tx FIFO element. When the Put Index reaches the Get Index, Tx FIFO Full (MCAN_TXFQS.TFQF = ‘1’) is signalled. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

When a single message is added to the Tx FIFO, the transmission is requested by writing a ‘1’ to the TXBAR bit related to the Tx Buffer referenced by the Tx FIFO’s Put Index.

When multiple (n) messages are added to the Tx FIFO, they are written to n consecutive Tx Buffers starting with the Put Index. The transmissions are then requested via MCAN_TXBAR. The Put Index is then cyclically incremented by n. The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level.

When a transmission request for the Tx Buffer referenced by the Get Index is cancelled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level is recalculated. When transmission cancellation is applied to any other Tx Buffer, the Get Index and the FIFO Free Level remain unchanged.

A Tx FIFO element allocates Element Size 32-bit words in the Message RAM (see Table 50-8). Therefore the start address of the next available (free) Tx FIFO Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS.TFQPI (0…31) × Element Size to the Tx Buffer Start Address MCAN_TXBC.TBSA.

50.5.5.4 Tx Queue

Tx Queue operation is configured by programming MCAN_TXBC.TFQM to ‘1’. Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). In case that multiple Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New messages have to be written to the Tx Buffer referenced by the Put Index MCAN_TXFQS.TFQPI. An Add Request cyclically increments the Put Index to the next free Tx Buffer. In case that the Tx Queue is full (MCAN_TXFQS.TFQF = '1'), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been cancelled.

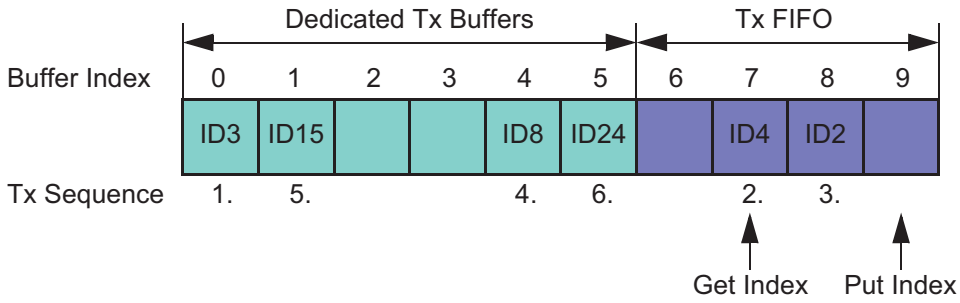
The application may use register MCAN_TXBRP instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates Element Size 32-bit words in the Message RAM (see Table 50-8). Therefore the start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS.TFQPI (0…31) × Element Size to the Tx Buffer Start Address MCAN_TXBC.TBSA.

50.5.5.5 Mixed Dedicated Tx Buffers / Tx FIFO

In this case the Tx Buffers section in the Message RAM is subdivided into a set of dedicated Tx Buffers and a Tx FIFO. The number of dedicated Tx Buffers is configured by MCAN_TXBC.NDTB. The number of Tx Buffers assigned to the Tx FIFO is configured by MCAN_TXBC.TFQS. In case MCAN_TXBC.TFQS is programmed to zero, only dedicated Tx Buffers are used.

Figure 50-10. Example of Mixed Configuration Dedicated Tx Buffers / Tx FIFO



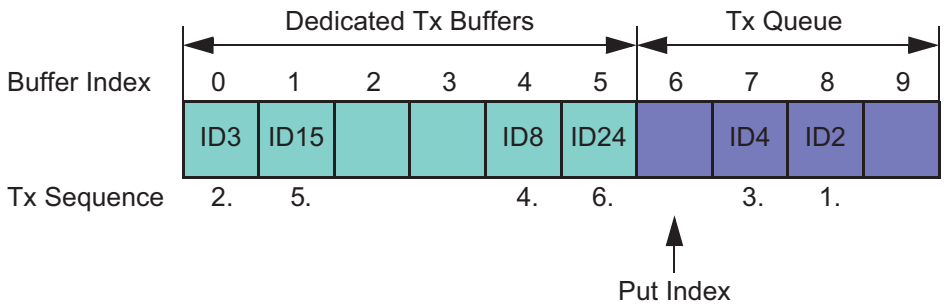
Tx prioritization:

- Scan dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by MCAN_TXFS.TFGI)
- Buffer with lowest Message ID gets highest priority and is transmitted next

50.5.5.6 Mixed Dedicated Tx Buffers / Tx Queue

In this case the Tx Buffers section in the Message RAM is subdivided into a set of dedicated Tx Buffers and a Tx Queue. The number of dedicated Tx Buffers is configured by MCAN_TXBC.NDTB. The number of Tx Queue Buffers is configured by MCAN_TXBC.TFQS. In case MCAN_TXBC.TFQS is programmed to zero, only dedicated Tx Buffers are used.

Figure 50-11. Example of Mixed Configuration Dedicated Tx Buffers / Tx Queue



Tx prioritization:

- Scan all Tx Buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next

50.5.5.7 Transmit Cancellation

The MCAN supports transmit cancellation. This feature is especially intended for gateway applications and AUTOSAR-based applications. To cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer, the processor has to write a '1' to the corresponding bit position (=number of Tx Buffer) of register MCAN_TXBCR. Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signalled by setting the corresponding bit of register MCAN_TXBCF to '1'.

In case a transmit cancellation is requested while a transmission from a Tx Buffer is already ongoing, the corresponding TXBRP bit remains set as long as the transmission is in progress. If the transmission was successful, the corresponding MCAN_TXBTO and MCAN_TXBCF bits are set. If the transmission was not successful, it is not repeated and only the corresponding MCAN_TXBCF bit is set.

Note: In case a pending transmission is cancelled immediately before this transmission could have been started, there follows a short time window where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority than the second message in this node.

50.5.5.8 Tx Event Handling

To support Tx event handling the MCAN has implemented a Tx Event FIFO. After the MCAN has transmitted a message on the CAN bus, Message ID and timestamp are stored in a Tx Event FIFO element. To link a Tx event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

The Tx Event FIFO can be configured to a maximum of 32 elements. The Tx Event FIFO element is described in Section 50.5.4.4.

When a Tx Event FIFO full condition is signalled by IR.TEFF, no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index has been incremented. In case a Tx event occurs while the Tx Event FIFO is full, this event is discarded and interrupt flag MCAN_IR.TEFL is set.

To avoid a Tx Event FIFO overflow, the Tx Event FIFO watermark can be used. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by MCAN_TXEFC.EFWM, interrupt flag MCAN_IR.TEFW is set.

When reading from the Tx Event FIFO, two times the Tx Event FIFO Get Index MCAN_TXEFS.EFGI has to be added to the Tx Event FIFO start address MCAN_TXEFC.EFSA.

50.5.6 FIFO Acknowledge Handling

The Get Indices of Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index (see Section 50.6.29, Section 50.6.33, and Section 50.6.47). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level. There are two use cases:

When only a single element has been read from the FIFO (the one being pointed to by the Get Index), this Get Index value is written to the FIFO Acknowledge Index.

When a sequence of elements has been read from the FIFO, it is sufficient to write the FIFO Acknowledge Index only once at the end of that read sequence (value: Index of the last element read), to update the FIFO's Get Index.

Due to the fact that the processor has free access to the MCAN's Message RAM, special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This might be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the FIFO's Acknowledge Index should not be written because this would set the Get Index to a wrong position and also alters the FIFO's Fill Level. In this case some of the older FIFO elements would be lost.

Note: The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The MCAN does not check for erroneous values.

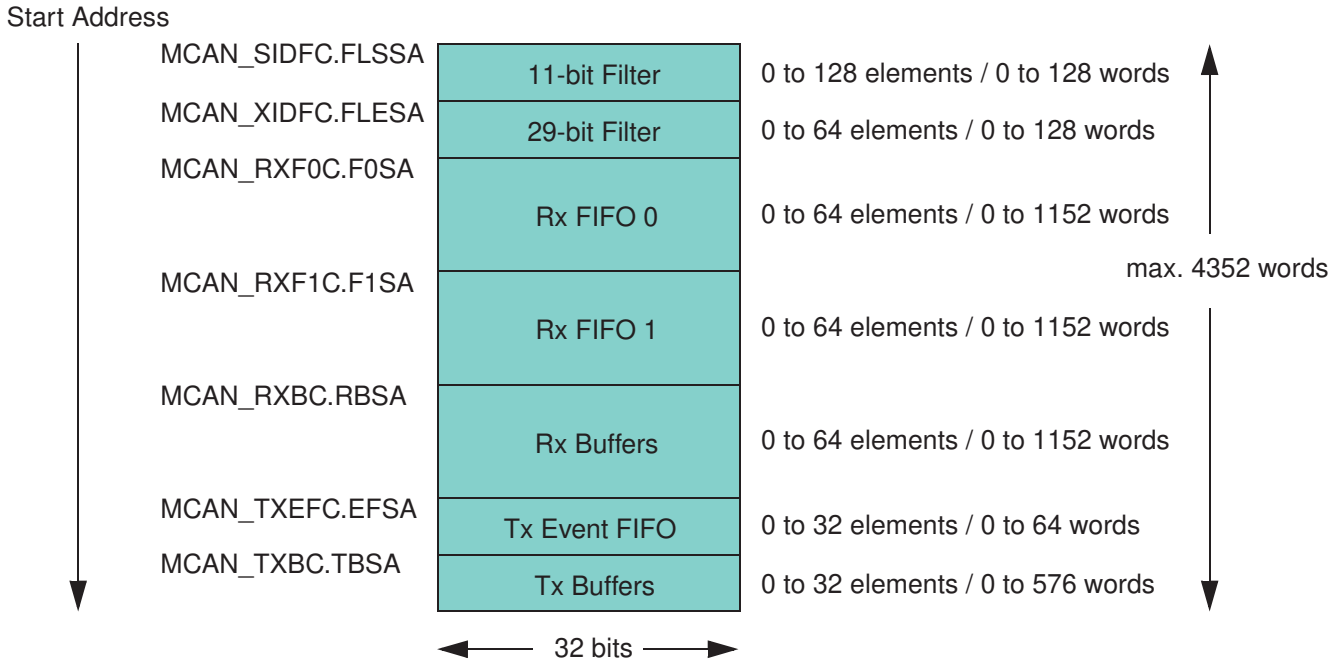
50.5.7 Message RAM

50.5.7.1 Message RAM Configuration

The Message RAM has a width of 32 bits. The MCAN module can be configured to allocate up to 4352 words in the Message RAM. It is not necessary to configure each of the sections listed in Figure 50-12, nor is there any restriction with respect to the sequence of the sections.

When operated in CAN FD mode, the required Message RAM size depends on the element size configured for Rx FIFO0, Rx FIFO1, Rx Buffers, and Tx Buffers via MCAN_RXESC.F0DS, MCAN_RXESC.F1DS, MCAN_RXESC.RBDS, and MCAN_TXESC.TBDS.

Figure 50-12. Message RAM Configuration



When the MCAN addresses the Message RAM, it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses; i.e., only bits 15 to 2 are evaluated, the two least significant bits are ignored.

Note: The MCAN does not check for erroneous configuration of the Message RAM. The configuration of the start addresses of the different sections and the number of elements of each section must be checked carefully to avoid falsification or loss of data.

50.5.7.2 Rx Buffer and FIFO Element

Up to 64 Rx Buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The structure of a Rx Buffer / FIFO element is shown in Table 50-9 below. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register MCAN_RXESC.

Table 50-9. Rx Buffer and FIFO Element

<table>
<tr><th></th><th colspan="4">31 … 24</th><th colspan="4">23 … 16</th><th>15 … 8</th><th>7 … 0</th></tr>
<tr><td>R0</td><td>ESI</td><td>XTD</td><td>RTR</td><td colspan="7">ID[28:0]</td></tr>
<tr><td>R1</td><td>ANMF</td><td colspan="3">FIDX[6:0]</td><td>–</td><td>FDF</td><td>BRS</td><td>DLC[3:0]</td><td colspan="2">RXTS[15:0]</td></tr>
<tr><td>R2</td><td colspan="4">DB3[7:0]</td><td colspan="4">DB2[7:0]</td><td>DB1[7:0]</td><td>DB0[7:0]</td></tr>
<tr><td>R3</td><td colspan="4">DB7[7:0]</td><td colspan="4">DB6[7:0]</td><td>DB5[7:0]</td><td>DB4[7:0]</td></tr>
<tr><td>…</td><td colspan="4">...</td><td colspan="4">...</td><td>...</td><td>...</td></tr>
<tr><td>Rn</td><td colspan="4">DBm[7:0]</td><td colspan="4">DBm-1[7:0]</td><td>DBm-2[7:0]</td><td>DBm-3[7:0]</td></tr>
</table>

- **R0 Bit 31 ESI: Error State Indicator**

0: Transmitting node is error active.

1: Transmitting node is error passive.

- **R0 Bit 30 XTD: Extended Identifier**

Signals to the processor whether the received frame has a standard or extended identifier.

0: 11-bit standard identifier.

1: 29-bit extended identifier.

- **R0 Bit 29 RTR: Remote Transmission Request**

Signals to the processor whether the received frame is a data frame or a remote frame.

0: Received frame is a data frame.

1: Received frame is a remote frame.

Note: There are no remote frames in CAN FD format. In case a CAN FD frame was received (FDF = 1), bit RTR reflects the state of the reserved bit r1.

- **R0 Bits 28:0 ID[28:0]: Identifier**

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

- **R1 Bit 31 ANMF: Accepted Non-matching Frame**

Acceptance of non-matching frames may be enabled via MCAN_GFC.ANFS and MCAN_GFC.ANFE.

0: Received frame matching filter index FIDX.

1: Received frame did not match any Rx filter element.

- **R1 Bits 30:24 FIDX[6:0]: Filter Index**

0-127: Index of matching Rx acceptance filter element (invalid if ANMF = ‘1’).
Range is 0 to MCAN_SIDFC.LSS - 1 resp. MCAN_XIDFC.LSE - 1.

- **R1 Bit 21 FDF: FD Format**

0: Standard frame format.

1: CAN FD frame format (new DLC-coding and CRC).

- **R1 Bit 20 BRS: Bit Rate Switch**

0: Frame received without bit rate switching.

1: Frame received with bit rate switching.

Note: Bits ESI, FDF, and BRS are only evaluated when CAN FD operation is enabled (MCAN_CCCR.FDOE = 1). Bit BRS is only evaluated when in addition MCAN_CCCR.BRSE = 1.

- **R1 Bits 19:16 DLC[3:0]: Data Length Code**

0-8: CAN + CAN FD: received frame has 0-8 data bytes.

9-15: CAN: received frame has 8 data bytes.

9-15: CAN FD: received frame has 12/16/20/24/32/48/64 data bytes.

- **R1 Bits 15:0 RXTS[15:0]: Rx Timestamp**

Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC.TCP.

- **R2 Bits 31:24 DB3[7:0]: Data Byte 3**
- **R2 Bits 23:16 DB2[7:0]: Data Byte 2**
- **R2 Bits 15:8 DB1[7:0]: Data Byte 1**
- **R2 Bits 7:0 DB0[7:0]: Data Byte 0**
- **R3 Bits 31:24 DB7[7:0]: Data Byte 7**
- **R3 Bits 23:16 DB6[7:0]: Data Byte 6**
- **R3 Bits 15:8 DB5[7:0]: Data Byte 5**
- **R3 Bits 7:0 DB4[7:0]: Data Byte 4**

... ...

- **Rn Bits 31:24 DBm[7:0]: Data Byte m**
- **Rn Bits 23:16 DBm-1[7:0]: Data Byte m-1**
- **Rn Bits 15:8 DBm-2[7:0]: Data Byte m-2**
- **Rn Bits 7:0 DBm-3[7:0]: Data Byte m-3**

Note: Depending on the configuration of the element size (MCAN_RXESC), between two and sixteen 32-bit words (Rn = 3 ..17) are used for storage of a CAN message's data field.

50.5.7.3 Tx Buffer Element

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO / Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO / Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler distinguishes between dedicated Tx Buffers and Tx FIFO / Tx Queue by evaluating the Tx Buffer configuration TXBC.TFQS and TXBC.NDTB. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register TXESC.

Table 50-10. Tx Buffer Element

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T0	ESI	XTD	RTR	ID[28:0]																												
T1	MM[7:0]								EFC	reserved	FDF	BRS	DLC[3:0]				reserved															
T2	DB3[7:0]								DB2[7:0]								DB1[7:0]								DB0[7:0]							
T3	DB7[7:0]								DB6[7:0]								DB5[7:0]								DB4[7:0]							
...	...								...								...								...							
Tn	DBm[7:0]								DBm-1[7:0]								DBm-2[7:0]								DBm-3[7:0]							

- **T0 Bit 30 ESI: Error State Indicator**

T0 Bit 31 ESI: Error State Indicator

0: ESI bit in CAN FD format depends only on error passive flag

1: ESI bit in CAN FD format transmitted recessive

Note: The ESI bit of the transmit buffer is or'ed with the error passive flag to decide the value of the ESI bit in the transmitted FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive. This feature can be used in gateway applications when a message from an error passive node is routed to another CAN network.

- **T0 Bit 30 XTD: Extended Identifier**

0: 11-bit standard identifier.

1: 29-bit extended identifier.

- **T0 Bit 29 RTR: Remote Transmission Request**

0: Transmit data frame.

1: Transmit remote frame.

Note: When RTR = 1, the MCAN transmits a remote frame according to ISO11898-1, even if MCAN_CCCR.FDOE enables the transmission in CAN FD format.

- **T0 Bits 28:0 ID[28:0]: Identifier**

Standard or extended identifier depending on bit XTD. A standard identifier has to be written to ID[28:18].

- **T1 Bits 31:24 MM[7:0]: Message Marker**

Written by processor during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.

- **T1 Bit 23 EFC: Event FIFO Control**

0: Do not store Tx events.

1: Store Tx events.

- **T1 Bit 21 FDF: FD Format**

0: Frame transmitted in Classic CAN format

1: Frame transmitted in CAN FD format

- **T1 Bit 20 BRS: Bit Rate Switching**

0: CAN FD frames transmitted without bit rate switching

1: CAN FD frames transmitted with bit rate switching

Note: Bits ESI, FDF, and BRS are only evaluated when CAN FD operation is enabled (MCAN_CCCR.FDOE = 1). Bit BRS is only evaluated when in addition MCAN_CCCR.BRSE = 1.

- **T1 Bits 19:16 DLC[3:0]: Data Length Code**

0-8: CAN + CAN FD: transmit frame has 0-8 data bytes.

9-15: CAN: transmit frame has 8 data bytes.

9-15: CAN FD: transmit frame has 12/16/20/24/32/48/64 data bytes.

- **T2 Bits 31:24 DB3[7:0]: Data Byte 3**
- **T2 Bits 23:16 DB2[7:0]: Data Byte 2**
- **T2 Bits 15:8 DB1[7:0]: Data Byte 1**
- **T2 Bits 7:0 DB0[7:0]: Data Byte 0**
- **T3 Bits 31:24 DB7[7:0]: Data Byte 7**
- **T3 Bits 23:16 DB6[7:0]: Data Byte 6**
- **T3 Bits 15:8 DB5[7:0]: Data Byte 5**
- **T3 Bits 7:0 DB4[7:0]: Data Byte 4**

... ...

- **Tn Bits 31:24 DBm[7:0]: Data Byte m**
- **Tn Bits 23:16 DBm-1[7:0]: Data Byte m-1**
- **Tn Bits 15:8 DBm-2[7:0]: Data Byte m-2**
- **Tn Bits 7:0 DBm-3[7:0]: Data Byte m-3**

Note: Depending on the configuration of the element size (MCAN_TXESC), between two and sixteen 32-bit words (Tn = 3 ..17) are used for storage of a CAN message's data field.

50.5.7.4 Tx Event FIFO Element

Each element stores information about transmitted messages. By reading the Tx Event FIFO the processor gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from register TXEFS.

Table 50-11. Tx Event FIFO Element

	31				24	23				16	15				8	7				0
E0	ESI	XTD	RTR		ID[28:0]															
E1	MM[7:0]					ET [1:0]		FDF	BRS	DLC[3:0]	TXTS[15:0]									

- **E0 Bit 31 ESI: Error State Indicator**

0: Transmitting node is error active.

1: Transmitting node is error passive.

- **E0 Bit 30 XTD: Extended Identifier**

0: 11-bit standard identifier.

1: 29-bit extended identifier.

- **E0 Bit 29 RTR: Remote Transmission Request**

0: Data frame transmitted.

1: Remote frame transmitted.

- **E0 Bits 28:0 ID[28:0]: Identifier**

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

- **E1 Bits 31:24 MM[7:0]: Message Marker**

Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status.

- **E1 Bit 23:22 ET[1:0]: Event Type**

Value	Description
0	Reserved
1	Tx event
2	Transmission in spite of cancellation (always set for transmissions in DAR mode)
3	Reserved

- **E1 Bit 21 FDF: FD Format**

0: Standard frame format.

1: CAN FD frame format (new DLC-coding and CRC).

- **E1 Bit 20 BRS: Bit Rate Switch**

0: Frame transmitted without bit rate switching.

1: Frame transmitted with bit rate switching.

- **E1 Bits 19:16 DLC[3:0]: Data Length Code**

0-8: CAN + CAN FD: frame with 0-8 data bytes transmitted.

9-15: CAN: frame with 8 data bytes transmitted.

9-15: CAN FD: frame with 12/16/20/24/32/48/64 data bytes transmitted

- **E1 Bits 15:0 TXTS[15:0]: Tx Timestamp**

Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC.TCP.

50.5.7.5 Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, its address is the Filter List Standard Start Address MCAN_SIDFC.FLSSA plus the index of the filter element (0…127).

Table 50-12. Standard Message ID Filter Element

	31	24	23	16	15	8	7	0
S0	SFT[1:0]	SFEC [2:0]	SFID1[10:0]		–		SFID2[10:0]	

- **Bits 31:30 SFT[1:0]: Standard Filter Type**

Value	Description
0	Range filter from SF1ID to SF2ID (SF2ID ≥ SF1ID)
1	Dual ID filter for SF1ID or SF2ID
2	Classic filter: SF1ID = filter, SF2ID = mask
3	Reserved

- **Bit 29:27 SFEC[2:0]: Standard Filter Element Configuration**

All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = “100”, “101”, or “110” a match sets interrupt flag MCAN_IR.HPM and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match.

Value	Description
0	Disable filter element
1	Store in Rx FIFO 0 if filter matches
2	Store in Rx FIFO 1 if filter matches
3	Reject ID if filter matches
4	Set priority if filter matches
5	Set priority and store in FIFO 0 if filter matches
6	Set priority and store in FIFO 1 if filter matches
7	Store into Rx Buffer or as debug message, configuration of SFT[1:0] ignored

- **Bits 26:16 SFID1[10:0]: Standard Filter ID 1**

First ID of standard ID filter element.

When filtering for Rx Buffers or for debug messages this field defines the ID of a standard message to be stored. The received identifiers must match exactly, no masking mechanism is used.

- **Bits 10:0 SFID2[10:0]: Standard Filter ID 2**

This field has a different meaning depending on the configuration of SFEC:

- SFEC = “001”...“110”–Second ID of standard ID filter element
- SFEC = “111”–Filter for Rx Buffers or for debug messages

SFID2[10:9] decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.

Value	Description
0	Store message in a Rx buffer
1	Debug Message A
2	Debug Message B
3	Debug Message C

SFID2[5:0] defines the index of the dedicated Rx Buffer element to which a matching message is stored.

50.5.7.6 Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address MCAN_XIDFC.FLESA plus two times the index of the filter element (0…63).

Table 50-13. Extended Message ID Filter Element

	31 … 24	23 … 16	15 … 8	7 … 0
F0	EFEC [2:0]	EFID1[28:0]		
F1	EFT[1:0]	–	EFID2[28:0]	

- **F0 Bit 31:29 EFEC[2:0]: Extended Filter Element Configuration**

All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = “100”, “101”, or “110”, a match sets the interrupt flag MCAN_IR.HPM and, if enabled, an interrupt is generated. In this case, register MCAN_HPMS is updated with the status of the priority match.

Value	Description
0	Disable filter element
1	Store in Rx FIFO 0 if filter matches
2	Store in Rx FIFO 1 if filter matches
3	Reject ID if filter matches
4	Set priority if filter matches
5	Set priority and store in FIFO 0 if filter matches
6	Set priority and store in FIFO 1 if filter matches
7	Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored

- **F0 Bits 28:0 EFID1[28:0]: Extended Filter ID 1**

First ID of extended ID filter element.

When filtering for Rx Buffers or for debug messages this field defines the ID of an extended message to be stored. The received identifiers must match exactly, only MCAN_XIDAM masking mechanism (see Extended Message ID Filtering) is used.

- **F1 Bits 31:30 EFT[1:0]: Extended Filter Type**

Value	Description
0	Range filter from EF1ID to EF2ID (EF2ID ≥ EF1ID)
1	Dual ID filter for EF1ID or EF2ID
2	Classic filter: EF1ID = filter, EF2ID = mask
3	Range filter from EF1ID to EF2ID (EF2ID ≥ EF1ID), MCAN_XIDAM mask not applied

- **F1 Bits 28:0 EFID2[28:0]: Extended Filter ID 2**

This field has a different meaning depending on the configuration of EFEC:

- EFEC = “001”...“110”–Second ID of extended ID filter element
- EFEC = “111”–Filter for Rx Buffers or for debug messages

EFID2[10:9] decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.

Value	Description
0	Store message in a Rx buffer
1	Debug Message A
2	Debug Message B
3	Debug Message C

EFID2[5:0] defines the index of the dedicated Rx Buffer element to which a matching message is stored.

50.5.8 Hardware Reset Description

After hardware reset, the registers of the MCAN hold the reset values listed in Table 50-14. Additionally the Bus_Off state is reset and the output CANTX is set to recessive (HIGH). The value 0x0001 (MCAN_CCCR.INIT = ‘1’) in the CC Control register enables software initialization. The MCAN does not influence the CAN bus until the processor resets MCAN_CCCR.INIT to ‘0’.

50.5.9 Access to Reserved Register Addresses

In case the application software accesses one of the reserved addresses in the MCAN register map (read or write access), interrupt flag MCAN_IR.ARA is set and, if enabled, the selected interrupt line is risen.

50.6 Controller Area Network (MCAN) User Interface

Table 50-14. Register Mapping

Offset	Register	Name	Access	Reset
0x00	Core Release Register	MCAN_CREL	Read-only	0xrrrddddd[1]
0x04	Endian Register	MCAN_ENDN	Read-only	0x87654321
0x08	Customer Register	MCAN_CUST	Read/Write	0
0x0C	Data Bit Timing and Prescaler Register	MCAN_DBTP	Read/Write	0x00000A33
0x10	Test Register	MCAN_TEST	Read/Write	0x000000x0[2]
0x14	RAM Watchdog Register	MCAN_RWD	Read/Write	0x00000000
0x18	CC Control Register	MCAN_CCCR	Read/Write	0x00000001
0x1C	Nominal Bit Timing and Prescaler Register	MCAN_NBTP	Read/Write	0x06000A03
0x20	Timestamp Counter Configuration Register	MCAN_TSCC	Read/Write	0x00000000
0x24	Timestamp Counter Value Register	MCAN_TSCV	Read/Write	0x00000000
0x28	Timeout Counter Configuration Register	MCAN_TOCC	Read/Write	0xFFFF0000
0x2C	Timeout Counter Value Register	MCAN_TOCV	Read/Write	0x0000FFFF
0x30–0x3C	Reserved	–	–	–
0x40	Error Counter Register	MCAN_ECR	Read-only	0x00000000
0x44	Protocol Status Register	MCAN_PSR	Read-only	0x00000707
0x48	Transmit Delay Compensation Register	MCAN_TDCR	Read/Write	0x00000000
0x4C	Reserved	–	–	–
0x50	Interrupt Register	MCAN_IR	Read/Write	0x00000000
0x54	Interrupt Enable Register	MCAN_IE	Read/Write	0x00000000
0x58	Interrupt Line Select Register	MCAN_ILS	Read/Write	0x00000000
0x5C	Interrupt Line Enable Register	MCAN_ILE	Read/Write	0x00000000
0x60–0x7C	Reserved	–	–	–
0x80	Global Filter Configuration Register	MCAN_GFC	Read/Write	0x00000000
0x84	Standard ID Filter Configuration Register	MCAN_SIDFC	Read/Write	0x00000000
0x88	Extended ID Filter Configuration Register	MCAN_XIDFC	Read/Write	0x00000000
0x8C	Reserved	–	–	–
0x90	Extended ID AND Mask Register	MCAN_XIDAM	Read/Write	0x1FFFFFFF
0x94	High Priority Message Status Register	MCAN_HPMS	Read-only	0x00000000
0x98	New Data 1 Register	MCAN_NDAT1	Read/Write	0x00000000
0x9C	New Data 2 Register	MCAN_NDAT2	Read/Write	0x00000000
0xA0	Receive FIFO 0 Configuration Register	MCAN_RXF0C	Read/Write	0x00000000
0xA4	Receive FIFO 0 Status Register	MCAN_RXF0S	Read-only	0x00000000
0xA8	Receive FIFO 0 Acknowledge Register	MCAN_RXF0A	Read/Write	0x00000000
0xAC	Receive Rx Buffer Configuration Register	MCAN_RXBC	Read/Write	0x00000000
0xB0	Receive FIFO 1 Configuration Register	MCAN_RXF1C	Read/Write	0x00000000

Table 50-14. Register Mapping (Continued)

Offset	Register	Name	Access	Reset
0xB4	Receive FIFO 1 Status Register	MCAN_RXF1S	Read-only	0x00000000
0xB8	Receive FIFO 1 Acknowledge Register	MCAN_RXF1A	Read/Write	0x00000000
0xBC	Receive Buffer / FIFO Element Size Configuration Register	MCAN_RXESC	Read/Write	0x00000000
0xC0	Transmit Buffer Configuration Register	MCAN_TXBC	Read/Write	0x00000000
0xC4	Transmit FIFO/Queue Status Register	MCAN_TXFQS	Read-only	0x00000000
0xC8	Transmit Buffer Element Size Configuration Register	MCAN_TXESC	Read/Write	0x00000000
0xCC	Transmit Buffer Request Pending Register	MCAN_TXBRP	Read-only	0x00000000
0xD0	Transmit Buffer Add Request Register	MCAN_TXBAR	Read/Write	0x00000000
0xD4	Transmit Buffer Cancellation Request Register	MCAN_TXBCR	Read/Write	0x00000000
0xD8	Transmit Buffer Transmission Occurred Register	MCAN_TXBTO	Read-only	0x00000000
0xDC	Transmit Buffer Cancellation Finished Register	MCAN_TXBCF	Read-only	0x00000000
0xE0	Transmit Buffer Transmission Interrupt Enable Register	MCAN_TXBTIE	Read/Write	0x00000000
0xE4	Transmit Buffer Cancellation Finished Interrupt Enable Register	MCAN_TXBCIE	Read/Write	0x00000000
0xE8–0xEC	Reserved	–	–	–
0xF0	Transmit Event FIFO Configuration Register	MCAN_TXEFC	Read/Write	0x00000000
0xF4	Transmit Event FIFO Status Register	MCAN_TXEFS	Read-only	0x00000000
0xF8	Transmit Event FIFO Acknowledge Register	MCAN_TXEFA	Read/Write	0x00000000
0xFC	Reserved	–	–	–

Notes:
1. Due to clock domain crossing, there is a delay between when a register bit or field is written and when the related status register bits are updated.
2. The reset value for bit 7, MCAN_TEST.RX, is undefined.

50.6.1 MCAN Core Release Register

Name: MCAN_CREL

Address: 0xF8054000 (0), 0xFC050000 (1)

Access: Read-only

31	30	29	28	27	26	25	24
REL				STEP			
23	22	21	20	19	18	17	16
SUBSTEP				YEAR			
15	14	13	12	11	10	9	8
MON							
7	6	5	4	3	2	1	0
DAY							

- **DAY: Timestamp Day**

Two digits, BCD-coded. This field is set by generic parameter on MCAN synthesis.

- **MON: Timestamp Month**

Two digits, BCD-coded. This field is set by generic parameter on MCAN synthesis.

- **YEAR: Timestamp Year**

One digit, BCD-coded. This field is set by generic parameter on MCAN synthesis.

- **SUBSTEP: Sub-step of Core Release**

One digit, BCD-coded.

- **STEP: Step of Core Release**

One digit, BCD-coded.

- **REL: Core Release**

One digit, BCD-coded.

50.6.2 MCAN Endian Register

Name: MCAN_ENDN

Address: 0xF8054004 (0), 0xFC050004 (1)

Access: Read-only

31	30	29	28	27	26	25	24
ETV							
23	22	21	20	19	18	17	16
ETV							
15	14	13	12	11	10	9	8
ETV							
7	6	5	4	3	2	1	0
ETV							

- **ETV: Endianness Test Value**

The endianness test value is 0x87654321.

50.6.3 MCAN Customer Register

Name: MCAN_CUST

Address: 0xF8054008 (0), 0xFC050008 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
CSV							
23	22	21	20	19	18	17	16
CSV							
15	14	13	12	11	10	9	8
CSV							
7	6	5	4	3	2	1	0
CSV							

- **CSV: Customer-specific Value**

Customer-specific value.

50.6.4 MCAN Data Bit Timing and Prescaler Register

Name: MCAN_DBTP

Address: 0xF805400C (0), 0xFC05000C (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
TDC	–	–	DBRP				
15	14	13	12	11	10	9	8
–	–	–	DTSEG1				
7	6	5	4	3	2	1	0
DTSEG2				–	DSJW		

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

The CAN bit time may be programmed in the range of 4 to 25 time quanta. The CAN time quantum may be programmed in the range of 1 to 32 CAN core clock periods. t_q = (DBRP + 1) CAN core clock periods.

DTSEG1 is the sum of Prop_Seg and Phase_Seg1. DTSEG2 is Phase_Seg2.

Therefore the length of the bit time is (programmed values) [DTSEG1 + DTSEG2 + 3] t_q
or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q.

The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.

- **DSJW: Data (Re) Synchronization Jump Width**

The duration of a synchronization jump is t_q x (DSJW + 1).

- **DTSEG2: Data Time Segment After Sample Point**

The duration of time segment is t_q x (DTSEG2 + 1).

- **DTSEG1: Data Time Segment Before Sample Point**

0: Forbidden.

1 to 31: The duration of time segment is t_q x (DTSEG1 + 1).

- **DBRP: Data Bit Rate Prescaler**

The value by which the peripheral clock is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

- **TDC: Transceiver Delay Compensation**

0 (DISABLED): Transceiver Delay Compensation disabled.

1 (ENABLED): Transceiver Delay Compensation enabled.

Notes:
1. With a CAN core clock frequency of 8 MHz, the reset value of 0x00000A33 configures the MCAN for a fast bit rate of 500 kbit/s.
2. The bit rate configured for the CAN FD data phase via MCAN_DBTP must be higher than or equal to the bit rate configured for the arbitration phase via MCAN_NBTP.

50.6.5 MCAN Test Register

Name: MCAN_TEST

Address: 0xF8054010 (0), 0xFC050010 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
RX	TX		LBCK	–	–	–	–

Write access to the Test Register has to be enabled by setting bit MCAN_CCCR.TEST to ‘1’.

All MCAN Test Register functions are set to their reset values when bit MCAN_CCCR.TEST is cleared.

Loop Back mode and software control of pin CANTX are hardware test modes. Programming of TX ≠ 0 disturbs the message transfer on the CAN bus.

- **LBCK: Loop Back Mode (read/write)**

0 (DISABLED): Reset value. Loop Back mode is disabled.

1 (ENABLED): Loop Back mode is enabled (see Section 50.5.1.9).

- **TX: Control of Transmit Pin (read/write)**

Value	Name	Description
0	RESET	Reset value, CANTX controlled by the CAN Core, updated at the end of the CAN bit time.
1	SAMPLE_POINT_MONITORING	Sample Point can be monitored at pin CANTX.
2	DOMINANT	Dominant (‘0’) level at pin CANTX.
3	RECESSIVE	Recessive (‘1’) at pin CANTX.

- **RX: Receive Pin (read-only)**

Monitors the actual value of pin CANRX.

0: The CAN bus is dominant (CANRX = ‘0’).

1: The CAN bus is recessive (CANRX = ‘1’).

50.6.6 MCAN RAM Watchdog Register

Name: MCAN_RWD

Address: 0xF8054014 (0), 0xFC050014 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
WDV							
7	6	5	4	3	2	1	0
WDC							

The RAM Watchdog monitors the Message RAM response time. A Message RAM access via the MCAN's Generic Master Interface starts the Message RAM Watchdog Counter with the value configured by MCAN_RWD.WDC. The counter is reloaded with MCAN_RWD.WDC when the Message RAM signals successful completion by activating its READY output. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt flag MCAN_IR.WDI is set. The RAM Watchdog Counter is clocked by the system bus clock (peripheral clock).

- **WDC: Watchdog Configuration (read/write)**

Start value of the Message RAM Watchdog Counter. The counter is disabled when WDC is cleared.

- **WDV: Watchdog Value (read-only)**

Watchdog Counter Value for the current message located in RAM.

50.6.7 MCAN CC Control Register

Name: MCAN_CCCR

Address: 0xF8054018 (0), 0xFC050018 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	TXP	EFBI	PXHD	–	–	BRSE	FDOE
7	6	5	4	3	2	1	0
TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT

- **INIT: Initialization (read/write)**

0 (DISABLED): Normal operation.

1 (ENABLED): Initialization is started.

Note: Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to INIT can be read back. Therefore the programmer has to ensure that the previous value written to INIT has been accepted by reading INIT before setting INIT to a new value.

- **CCE: Configuration Change Enable (read/write, write protection)**

0 (PROTECTED): The processor has no write access to the protected configuration registers.

1 (CONFIGURABLE): The processor has write access to the protected configuration registers (while MCAN_CCCR.INIT = '1').

- **ASM: Restricted Operation Mode (read/write, write protection against '1')**

For a description of the Restricted Operation mode see Section 50.5.1.5.

0 (NORMAL): Normal CAN operation.

1 (RESTRICTED): Restricted Operation mode active.

- **CSA: Clock Stop Acknowledge (read-only)**

0: No clock stop acknowledged.

1: MCAN may be set in power down by stopping the peripheral clock and the CAN core clock.

- **CSR: Clock Stop Request (read/write)**

0 (NO_CLOCK_STOP): No clock stop is requested.

1 (CLOCK_STOP): Clock stop requested. When clock stop is requested, first INIT and then CSA will be set after all pending transfer requests have been completed and the CAN bus reached idle.

- **MON: Bus Monitoring Mode (read/write, write protection against '1')**

0 (DISABLED): Bus Monitoring mode is disabled.

1 (ENABLED): Bus Monitoring mode is enabled.

- **DAR: Disable Automatic Retransmission (read/write, write protection)**

0 (AUTO_RETX): Automatic retransmission of messages not transmitted successfully enabled.

1 (NO_AUTO_RETX): Automatic retransmission disabled.

- **TEST: Test Mode Enable (read/write, write protection against ‘1’)**

0 (DISABLED): Normal operation, MCAN_TEST register holds reset values.

1 (ENABLED): Test mode, write access to MCAN_TEST register enabled.

- **FDOE: CAN FD Operation Enable (read/write, write protection)**

0 (DISABLED): FD operation disabled.

1 (ENABLED): FD operation enabled.

- **BRSE: Bit Rate Switching Enable (read/write, write protection)**

0 (DISABLED): Bit rate switching for transmissions disabled.

1 (ENABLED): Bit rate switching for transmissions enabled.

- **PXHD: Protocol Exception Event Handling (read/write, write protection)**

0: Protocol exception handling enabled.

1: Protocol exception handling disabled.

- **EFBI: Edge Filtering during Bus Integration (read/write, write protection)**

0: Edge filtering is disabled.

1: Edge filtering is enabled. Two consecutive dominant tq required to detect an edge for hard synchronization.

- **TXP: Transmit Pause (read/write, write protection)**

If this bit is set, the MCAN pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame (see Section 50.5.5).

0: Transmit pause disabled.

1: Transmit pause enabled.

50.6.8 MCAN Nominal Bit Timing and Prescaler Register

Name: MCAN_NBTP

Address: 0xF805401C (0), 0xFC05001C (1)

Access: Read/Write

31	30	29	28	27	26	25	24
NSJW							NBRP
23	22	21	20	19	18	17	16
NBRP							
15	14	13	12	11	10	9	8
NTSEG1							
7	6	5	4	3	2	1	0
–	NTSEG2						

This register can only be written if the bits CCE and INIT are set in MCAN_CCCR.

The CAN bit time may be programmed in the range of 4 to 385 time quanta. The CAN time quantum may be programmed in the range of 1 to 512 CAN core clock periods. $t_q = t_{core\ clock} \times (NBRP + 1)$.

NTSEG1 is the sum of Prop_Seg and Phase_Seg1. NTSEG2 is Phase_Seg2.

Therefore the length of the bit time is (programmed values) [NTSEG1 + NTSEG2 + 3] t_q
or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q.

The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.

- **NTSEG2: Nominal Time Segment After Sample Point**

0 to 127: The duration of time segment is $t_q \times (NTSEG2 + 1)$.

- **NTSEG1: Nominal Time Segment Before Sample Point**

0: Forbidden.

1 to 255: The duration of time segment is $t_q \times (NTSEG1 + 1)$.

- **NBRP: Nominal Bit Rate Prescaler**

0 to 511: The value by which the oscillator frequency is divided for generating the CAN time quanta. The CAN time is built up from a multiple of this quanta. CAN time quantum (tq) = $t_{core\ clock} \times (NBRP + 1)$

Note: With a CAN core clock frequency of 8 MHz, the reset value of 0x06000A03 configures the MCAN for a bit rate of 500 kbit/s.

- **NSJW: Nominal (Re) Synchronization Jump Width**

0 to 127: The duration of a synchronization jump is $t_q \times (NSJW + 1)$.

50.6.9 MCAN Timestamp Counter Configuration Register

Name: MCAN_TSCC

Address: 0xF8054020 (0), 0xFC050020 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	TCP			
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	TSS	

For a description of the Timestamp Counter see Section 50.5.2.

- **TSS: Timestamp Select**

Value	Name	Description
0	ALWAYS_0	Timestamp counter value always 0x0000
1	TCP_INC	Timestamp counter value incremented according to TCP
2	EXT_TIMESTAMP	External timestamp counter value used
3	ALWAYS_0	Timestamp counter value always 0x0000

- **TCP: Timestamp Counter Prescaler**

Configures the timestamp and timeout counters time unit in multiples of CAN bit times [1…16]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Note: With CAN FD, an external counter is required for timestamp generation (TSS = 2).

50.6.10 MCAN Timestamp Counter Value Register

Name: MCAN_TSCV

Address: 0xF8054024 (0), 0xFC050024 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
TSC							
7	6	5	4	3	2	1	0
TSC							

- **TSC: Timestamp Counter (cleared on write)**

The internal/external Timestamp Counter value is captured on start of frame (both Receive and Transmit). When MCAN_TSCC.TSS = 1, the Timestamp Counter is incremented in multiples of CAN bit times [1…16] depending on the configuration of MCAN_TSCC.TCP. A wrap around sets interrupt flag MCAN_IR.TSW. Write access resets the counter to zero.

When MCAN_TSCC.TSS = 2, TSC reflects the external Timestamp Counter value. Thus a write access has no impact.

Note: A “wrap around” is a change of the Timestamp Counter value from non-zero to zero not caused by write access to MCAN_TSCV.

50.6.11 MCAN Timeout Counter Configuration Register

Name: MCAN_TOCC

Address: 0xF8054028 (0), 0xFC050028 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
TOP							
23	22	21	20	19	18	17	16
TOP							
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	TOS		ETOC

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

For a description of the Timeout Counter, see Section 50.5.3.

- **ETOC: Enable Timeout Counter**

0 (NO_TIMEOUT): Timeout Counter disabled.

1 (TOS_CONTROLLED): Timeout Counter enabled.

For use of timeout function with CAN FD, see Section 50.5.3.

- **TOS: Timeout Select**

When operating in Continuous mode, a write to MCAN_TOCV presets the counter to the value configured by MCAN_TOCC.TOP and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by MCAN_TOCC.TOP. Down-counting is started when the first FIFO element is stored.

Value	Name	Description
0	CONTINUOUS	Continuous operation
1	TX_EV_TIMEOUT	Timeout controlled by Tx Event FIFO
2	RX0_EV_TIMEOUT	Timeout controlled by Receive FIFO 0
3	RX1_EV_TIMEOUT	Timeout controlled by Receive FIFO 1

- **TOP: Timeout Period**

Start value of the Timeout Counter (down-counter). Configures the Timeout Period.

50.6.12 MCAN Timeout Counter Value Register

Name: MCAN_TOCV

Address: 0xF805402C (0), 0xFC05002C (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
TOC							
7	6	5	4	3	2	1	0
TOC							

- **TOC: Timeout Counter (cleared on write)**

The Timeout Counter is decremented in multiples of CAN bit times [1…16] depending on the configuration of MCAN_TSCC.TCP. When decremented to zero, interrupt flag MCAN_IR.TOO is set and the Timeout Counter is stopped. Start and reset/restart conditions are configured via MCAN_TOCC.TOS.

50.6.13 MCAN Error Counter Register

Name: MCAN_ECR

Address: 0xF8054040 (0), 0xFC050040 (1)

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
CEL							
15	14	13	12	11	10	9	8
RP	REC						
7	6	5	4	3	2	1	0
TEC							

- **TEC: Transmit Error Counter**

Actual state of the Transmit Error Counter, values between 0 and 255.

- **REC: Receive Error Counter**

Actual state of the Receive Error Counter, values between 0 and 127.

- **RP: Receive Error Passive**

0: The Receive Error Counter is below the error passive level of 128.

1: The Receive Error Counter has reached the error passive level of 128.

- **CEL: CAN Error Logging (cleared on read)**

The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or the Receive Error Counter to be incremented. It is reset by read access to CEL. The counter stops at 0xFF; the next increment of TEC or REC sets interrupt flag IR.ELO.

Note: When MCAN_CCCR.ASM is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented.

50.6.14 MCAN Protocol Status Register

Name: MCAN_PSR

Address: 0xF8054044 (0), 0xFC050044 (1)

Access: Read-only/

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	TDCV						
15	14	13	12	11	10	9	8
–	PXE	RFDF	RBRS	RESI	DLEC		
7	6	5	4	3	2	1	0
BO	EW	EP	ACT		LEC		

- **LEC: Last Error Code (set to 111 on read)**

The LEC indicates the type of the last error to occur on the CAN bus. This field is cleared when a message has been transferred (reception or transmission) without error.

Value	Name	Description
0	NO_ERROR	No error occurred since LEC has been reset by successful reception or transmission.
1	STUFF_ERROR	More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.
2	FORM_ERROR	A fixed format part of a received frame has the wrong format.
3	ACK_ERROR	The message transmitted by the MCAN was not acknowledged by another node.
4	BIT1_ERROR	During transmission of a message (with the exception of the arbitration field), the device tried to send a recessive level (bit of logical value ‘1’), but the monitored bus value was dominant.
5	BIT0_ERROR	During transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device tried to send a dominant level (data or identifier bit logical value ‘0’), but the monitored bus value was recessive. During Bus_Off recovery, this status is set each time a sequence of 11 recessive bits has been monitored. This enables the processor to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed).
6	CRC_ERROR	The CRC check sum of a received message was incorrect. The CRC of an incoming message does not match the CRC calculated from the received data.
7	NO_CHANGE	Any read access to the Protocol Status Register reinitializes the LEC to ‘7’. When the LEC shows value ‘7’, no CAN bus event was detected since the last processor read access to the Protocol Status Register.

- **ACT: Activity**

Monitors the CAN communication state of the CAN module.

Value	Name	Description
0	SYNCHRONIZING	Node is synchronizing on CAN communication
1	IDLE	Node is neither receiver nor transmitter
2	RECEIVER	Node is operating as receiver
3	TRANSMITTER	Node is operating as transmitter

- **EP: Error Passive**

0: The MCAN is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected.

1: The MCAN is in the Error_Passive state.

- **EW: Warning Status**

0: Both error counters are below the Error_Warning limit of 96.

1: At least one of error counter has reached the Error_Warning limit of 96.

- **BO: Bus_Off Status**

0: The MCAN is not Bus_Off.

1: The MCAN is in Bus_Off state.

- **DLEC: Data Phase Last Error Code (set to 111 on read)**

Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for LEC. This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error.

- **RESI: ESI Flag of Last Received CAN FD Message (cleared on read)**

This bit is set together with RFDF, independently from acceptance filtering.

0: Last received CAN FD message did not have its ESI flag set.

1: Last received CAN FD message had its ESI flag set.

- **RBRS: BRS Flag of Last Received CAN FD Message (cleared on read)**

This bit is set together with RFDF, independently from acceptance filtering.

0: Last received CAN FD message did not have its BRS flag set.

1: Last received CAN FD message had its BRS flag set.

- **RFDF: Received a CAN FD Message (cleared on read)**

This bit is set independently from acceptance filtering.

0: Since this bit was reset by the CPU, no CAN FD message has been received

1: Message in CAN FD format with FDF flag set has been received

- **PXE: Protocol Exception Event (cleared on read)**

0: No protocol exception event occurred since last read access

1: Protocol exception event occurred

- **TDCV: Transceiver Delay Compensation Value**

0 to 127: Position of the secondary sample point, in CAN core clock periods, defined by the sum of the measured delay from CANTX to CANRX and MCAN_TDCR.TDCO.

50.6.15 MCAN Transmitter Delay Compensation Register

Name: MCAN_TDCR

Address: 0xF8054048 (0), 0xFC050048 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	TDCO						
7	6	5	4	3	2	1	0
–	TDCF						

- **TDCF: Transmitter Delay Compensation Filter**

0 to 127: defines the minimum value for the SSP position, in CAN core clock periods. Dominant edges on CANRX that would result in an earlier SSP position are ignored for transmitter delay measurement. The feature is enabled when TDCF is configured to a value greater than TDCO.

- **TDCO: Transmitter Delay Compensation Offset**

0 to 127: Offset value, in CAN core clock periods, defining the distance between the measured delay from CANTX to CANRX and the secondary sample point.

50.6.16 MCAN Interrupt Register

Name: MCAN_IR

Address: 0xF8054050 (0), 0xFC050050 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	ARA	PED	PEA	WDI	BO	EW
23	22	21	20	19	18	17	16
EP	ELO	–	–	DRX	TOO	MRAF	TSW
15	14	13	12	11	10	9	8
TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
7	6	5	4	3	2	1	0
RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N

The flags are set when one of the listed conditions is detected (edge-sensitive). The flags remain set until the processor clears them. A flag is cleared by writing a ‘1’ to the corresponding bit position. Writing a ‘0’ has no effect. A hard reset will clear the register. The configuration of IE controls whether an interrupt is generated. The configuration of ILS controls on which interrupt line an interrupt is signalled.

- **RF0N: Receive FIFO 0 New Message**

0: No new message written to Receive FIFO 0.

1: New message written to Receive FIFO 0.

- **RF0W: Receive FIFO 0 Watermark Reached**

0: Receive FIFO 0 fill level below watermark.

1: Receive FIFO 0 fill level reached watermark.

- **RF0F: Receive FIFO 0 Full**

0: Receive FIFO 0 not full.

1: Receive FIFO 0 full.

- **RF0L: Receive FIFO 0 Message Lost**

0: No Receive FIFO 0 message lost.

1: Receive FIFO 0 message lost, also set after write attempt to Receive FIFO 0 of size zero.

- **RF1N: Receive FIFO 1 New Message**

0: No new message written to Receive FIFO 1.

1: New message written to Receive FIFO 1.

- **RF1W: Receive FIFO 1 Watermark Reached**

0: Receive FIFO 1 fill level below watermark.

1: Receive FIFO 1 fill level reached watermark.

- **RF1F: Receive FIFO 1 Full**

0: Receive FIFO 1 not full.

1: Receive FIFO 1 full.

- **RF1L: Receive FIFO 1 Message Lost**

0: No Receive FIFO 1 message lost.

1: Receive FIFO 1 message lost, also set after write attempt to Receive FIFO 1 of size zero.

- **HPM: High Priority Message**

0: No high priority message received.

1: High priority message received.

- **TC: Transmission Completed**

0: No transmission completed.

1: Transmission completed.

- **TCF: Transmission Cancellation Finished**

0: No transmission cancellation finished.

1: Transmission cancellation finished.

- **TFE: Tx FIFO Empty**

0: Tx FIFO non-empty.

1: Tx FIFO empty.

- **TEFN: Tx Event FIFO New Entry**

0: Tx Event FIFO unchanged.

1: Tx Handler wrote Tx Event FIFO element.

- **TEFW: Tx Event FIFO Watermark Reached**

0: Tx Event FIFO fill level below watermark.

1: Tx Event FIFO fill level reached watermark.

- **TEFF: Tx Event FIFO Full**

0: Tx Event FIFO not full.

1: Tx Event FIFO full.

- **TEFL: Tx Event FIFO Element Lost**

0: No Tx Event FIFO element lost.

1: Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.

- **TSW: Timestamp Wraparound**

0: No timestamp counter wrap-around.

1: Timestamp counter wrapped around.

- **MRAF: Message RAM Access Failure**

The flag is set, when the Rx Handler

- has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message.
- was not able to write a message to the Message RAM. In this case message storage is aborted.

In both cases the FIFO put index is not updated resp. the New Data flag for a dedicated Receive Buffer is not set, a partly stored message is overwritten when the next message is stored to this location.

The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the MCAN is switched into Restricted Operation mode (see Section 50.5.1.5). To leave Restricted Operation mode, the processor has to reset MCAN_CCCR.ASM.

0: No Message RAM access failure occurred.

1: Message RAM access failure occurred.

- **TOO: Timeout Occurred**

0: No timeout.

1: Timeout reached.

- **DRX: Message stored to Dedicated Receive Buffer**

The flag is set whenever a received message has been stored into a dedicated Receive Buffer.

0: No Receive Buffer updated.

1: At least one received message stored into a Receive Buffer.

- **ELO: Error Logging Overflow**

0: CAN Error Logging Counter did not overflow.

1: Overflow of CAN Error Logging Counter occurred.

- **EP: Error Passive**

0: Error_Passive status unchanged.

1: Error_Passive status changed.

- **EW: Warning Status**

0: Error_Warning status unchanged.

1: Error_Warning status changed.

- **BO: Bus_Off Status**

0: Bus_Off status unchanged.

1: Bus_Off status changed.

- **WDI: Watchdog Interrupt**

0: No Message RAM Watchdog event occurred.

1: Message RAM Watchdog event due to missing READY.

- **PEA: Protocol Error in Arbitration Phase**

0: No protocol error in arbitration phase

1: Protocol error in arbitration phase detected (MCAN_PSR.LEC differs from 0 or 7)

- **PED: Protocol Error in Data Phase**

0: No protocol error in data phase

1: Protocol error in data phase detected (MCAN_PSR.DLEC differs from 0 or 7)

- **ARA: Access to Reserved Address**

0: No access to reserved address occurred

1: Access to reserved address occurred

50.6.17 MCAN Interrupt Enable Register

Name: MCAN_IE

Address: 0xF8054054 (0), 0xFC050054 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	ARAE	PEDE	PEAE	WDIE	BOE	EWE
23	22	21	20	19	18	17	16
EPE	ELOE	–	–	DRXE	TOOE	MRAFE	TSWE
15	14	13	12	11	10	9	8
TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
7	6	5	4	3	2	1	0
RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE

The following configuration values are valid for all listed bit names of this register:

0: Disables the corresponding interrupt.

1: Enables the corresponding interrupt.

- **RF0NE: Receive FIFO 0 New Message Interrupt Enable**
- **RF0WE: Receive FIFO 0 Watermark Reached Interrupt Enable**
- **RF0FE: Receive FIFO 0 Full Interrupt Enable**
- **RF0LE: Receive FIFO 0 Message Lost Interrupt Enable**
- **RF1NE: Receive FIFO 1 New Message Interrupt Enable**
- **RF1WE: Receive FIFO 1 Watermark Reached Interrupt Enable**
- **RF1FE: Receive FIFO 1 Full Interrupt Enable**
- **RF1LE: Receive FIFO 1 Message Lost Interrupt Enable**
- **HPME: High Priority Message Interrupt Enable**
- **TCE: Transmission Completed Interrupt Enable**
- **TCFE: Transmission Cancellation Finished Interrupt Enable**
- **TFEE: Tx FIFO Empty Interrupt Enable**
- **TEFNE: Tx Event FIFO New Entry Interrupt Enable**
- **TEFWE: Tx Event FIFO Watermark Reached Interrupt Enable**
- **TEFFE: Tx Event FIFO Full Interrupt Enable**
- **TEFLE: Tx Event FIFO Event Lost Interrupt Enable**
- **TSWE: Timestamp Wraparound Interrupt Enable**

- **MRAFE: Message RAM Access Failure Interrupt Enable**
- **TOOE: Timeout Occurred Interrupt Enable**
- **DRXE: Message stored to Dedicated Receive Buffer Interrupt Enable**
- **ELOE: Error Logging Overflow Interrupt Enable**
- **EPE: Error Passive Interrupt Enable**
- **EWE: Warning Status Interrupt Enable**
- **BOE: Bus_Off Status Interrupt Enable**
- **WDIE: Watchdog Interrupt Enable**
- **PEAE: Protocol Error in Arbitration Phase Enable**
- **PEDE: Protocol Error in Data Phase Enable**
- **ARAE: Access to Reserved Address Enable**

50.6.18 MCAN Interrupt Line Select Register

Name: MCAN_ILS

Address: 0xF8054058 (0), 0xFC050058 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	ARAL	PEDL	PEAL	WDIL	BOL	EWL
23	22	21	20	19	18	17	16
EPL	ELOL	–	–	DRXL	TOOL	MRAFL	TSWL
15	14	13	12	11	10	9	8
TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
7	6	5	4	3	2	1	0
RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from the Interrupt Register to one of the two module interrupt lines.

0: Interrupt assigned to interrupt line MCAN_INT0.

1: Interrupt assigned to interrupt line MCAN_INT1.

- **RF0NL: Receive FIFO 0 New Message Interrupt Line**
- **RF0WL: Receive FIFO 0 Watermark Reached Interrupt Line**
- **RF0FL: Receive FIFO 0 Full Interrupt Line**
- **RF0LL: Receive FIFO 0 Message Lost Interrupt Line**
- **RF1NL: Receive FIFO 1 New Message Interrupt Line**
- **RF1WL: Receive FIFO 1 Watermark Reached Interrupt Line**
- **RF1FL: Receive FIFO 1 Full Interrupt Line**
- **RF1LL: Receive FIFO 1 Message Lost Interrupt Line**
- **HPML: High Priority Message Interrupt Line**
- **TCL: Transmission Completed Interrupt Line**
- **TCFL: Transmission Cancellation Finished Interrupt Line**
- **TFEL: Tx FIFO Empty Interrupt Line**
- **TEFNL: Tx Event FIFO New Entry Interrupt Line**
- **TEFWL: Tx Event FIFO Watermark Reached Interrupt Line**
- **TEFFL: Tx Event FIFO Full Interrupt Line**
- **TEFLL: Tx Event FIFO Event Lost Interrupt Line**
- **TSWL: Timestamp Wraparound Interrupt Line**

- **MRAFL: Message RAM Access Failure Interrupt Line**
- **TOOL: Timeout Occurred Interrupt Line**
- **DRXL: Message stored to Dedicated Receive Buffer Interrupt Line**
- **ELOL: Error Logging Overflow Interrupt Line**
- **EPL: Error Passive Interrupt Line**
- **EWL: Warning Status Interrupt Line**
- **BOL: Bus_Off Status Interrupt Line**
- **WDIL: Watchdog Interrupt Line**
- **PEAL: Protocol Error in Arbitration Phase Line**
- **PEDL: Protocol Error in Data Phase Line**
- **ARAL: Access to Reserved Address Line**

50.6.19 MCAN Interrupt Line Enable

Name: MCAN_ILE

Address: 0xF805405C (0), 0xFC05005C (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	EINT1	EINT0

Each of the two interrupt lines to the processor can be enabled / disabled separately by programming bits EINT0 and EINT1.

- **EINT0: Enable Interrupt Line 0**

0: Interrupt line MCAN_INT0 disabled.

1: Interrupt line MCAN_INT0 enabled.

- **EINT1: Enable Interrupt Line 1**

0: Interrupt line MCAN_INT1 disabled.

1: Interrupt line MCAN_INT1 enabled.

50.6.20 MCAN Global Filter Configuration

Name: MCAN_GFC

Address: 0xF8054080 (0), 0xFC050080 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	ANFS		ANFE		RRFS	RRFE

Global settings for Message ID filtering. The Global Filter Configuration controls the filter path for standard and extended messages as described in Figure 50-5 and Figure 50-6.

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

- **RRFE: Reject Remote Frames Extended**

0 (FILTER): Filter remote frames with 29-bit extended IDs.

1 (REJECT): Reject all remote frames with 29-bit extended IDs.

- **RRFS: Reject Remote Frames Standard**

0 (FILTER): Filter remote frames with 11-bit standard IDs.

1 (REJECT): Reject all remote frames with 11-bit standard IDs.

- **ANFE: Accept Non-matching Frames Extended**

Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated.

Value	Name	Description
0	RX_FIFO_0	Accept in Rx FIFO 0
1	RX_FIFO_1	Accept in Rx FIFO 1
2-3	REJECTED	Message rejected

- **ANFS: Accept Non-matching Frames Standard**

Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated.

Value	Name	Description
0	RX_FIFO_0	Accept in Rx FIFO 0
1	RX_FIFO_1	Accept in Rx FIFO 1
2-3	REJECTED	Message rejected

50.6.21 MCAN Standard ID Filter Configuration

Name: MCAN_SIDFC

Address: 0xF8054084 (0), 0xFC050084 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
LSS							
15	14	13	12	11	10	9	8
FLSSA							
7	6	5	4	3	2	1	0
FLSSA						–	–

Settings for 11-bit standard Message ID filtering. The Standard ID Filter Configuration controls the filter path for standard messages as described in Figure 50-5.

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

- **FLSSA: Filter List Standard Start Address**

Start address of standard Message ID filter list (32-bit word address, see Figure 50-12).

Write FLSSA with the bits [15:2] of the 32-bit address.

- **LSS: List Size Standard**

0: No standard Message ID filter.

1-128: Number of standard Message ID filter elements.

>128: Values greater than 128 are interpreted as 128.

50.6.22 MCAN Extended ID Filter Configuration

Name: MCAN_XIDFC

Address: 0xF8054088 (0), 0xFC050088 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	LSE						
15	14	13	12	11	10	9	8
FLESA							
7	6	5	4	3	2	1	0
FLESA						–	–

Settings for 29-bit extended Message ID filtering. The Extended ID Filter Configuration controls the filter path for standard messages as described in Figure 50-6.

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

- **FLESA: Filter List Extended Start Address**

Start address of extended Message ID filter list (32-bit word address, see Figure 50-12).

Write FLESA with the bits [15:2] of the 32-bit address.

- **LSE: List Size Extended**

0: No extended Message ID filter.

1-64: Number of extended Message ID filter elements.

>64: Values greater than 64 are interpreted as 64.

50.6.23 MCAN Extended ID AND Mask

Name: MCAN_XIDAM

Address: 0xF8054090 (0), 0xFC050090 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	EIDM				
23	22	21	20	19	18	17	16
EIDM							
15	14	13	12	11	10	9	8
EIDM							
7	6	5	4	3	2	1	0
EIDM							

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

- **EIDM: Extended ID Mask**

For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active.

50.6.24 MCAN High Priority Message Status

Name: MCAN_HPMS

Address: 0xF8054094 (0), 0xFC050094 (1)

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
FLST	FIDX						
7	6	5	4	3	2	1	0
MSI		BIDX					

This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.

- **BIDX: Buffer Index**

Index of Receive FIFO element to which the message was stored. Only valid when MSI[1] = ‘1’.

- **MSI: Message Storage Indicator**

Value	Name	Description
0	NO_FIFO_SEL	No FIFO selected.
1	LOST	FIFO message lost.
2	FIFO_0	Message stored in FIFO 0.
3	FIFO_1	Message stored in FIFO 1.

- **FIDX: Filter Index**

Index of matching filter element. Range is 0 to MCAN_SIDFC.LSS - 1 resp. MCAN_XIDFC.LSE - 1.

- **FLST: Filter List**

Indicates the filter list of the matching filter element.

0: Standard filter list

1: Extended filter list

50.6.25 MCAN New Data 1

Name: MCAN_NDAT1

Address: 0xF8054098 (0), 0xFC050098 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
23	22	21	20	19	18	17	16
ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
15	14	13	12	11	10	9	8
ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
7	6	5	4	3	2	1	0
ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0

- **NDx: New Data**

The register holds the New Data flags of Receive Buffers 0 to 31. The flags are set when the respective Receive Buffer has been updated from a received frame. The flags remain set until the processor clears them. A flag is cleared by writing a ‘1’ to the corresponding bit position. Writing a ‘0’ has no effect. A hard reset will clear the register.

0: Receive Buffer not updated

1: Receive Buffer updated from new message

50.6.26 MCAN New Data 2

Name: MCAN_NDAT2

Address: 0xF805409C (0), 0xFC05009C (1)

Access: Read/Write

31	30	29	28	27	26	25	24
ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56
23	22	21	20	19	18	17	16
ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
15	14	13	12	11	10	9	8
ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40
7	6	5	4	3	2	1	0
ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32

- **NDx: New Data**

The register holds the New Data flags of Receive Buffers 32 to 63. The flags are set when the respective Receive Buffer has been updated from a received frame. The flags remain set until the processor clears them. A flag is cleared by writing a ‘1’ to the corresponding bit position. Writing a ‘0’ has no effect. A hard reset will clear the register.

0: Receive Buffer not updated.

1: Receive Buffer updated from new message.

50.6.27 MCAN Receive FIFO 0 Configuration

Name: MCAN_RXF0C

Address: 0xF80540A0 (0), 0xFC0500A0 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
F0OM	F0WM						
23	22	21	20	19	18	17	16
–	F0S						
15	14	13	12	11	10	9	8
F0SA							
7	6	5	4	3	2	1	0
F0SA						–	–

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

- **F0SA: Receive FIFO 0 Start Address**

Start address of Receive FIFO 0 in Message RAM (32-bit word address, see Figure 50-12).

Write F0SA with the bits [15:2] of the 32-bit address.

- **F0S: Receive FIFO 0 Size**

0: No Receive FIFO 0

1-64: Number of Receive FIFO 0 elements.

>64: Values greater than 64 are interpreted as 64.

The Receive FIFO 0 elements are indexed from 0 to F0S-1.

- **F0WM: Receive FIFO 0 Watermark**

0: Watermark interrupt disabled.

1-64: Level for Receive FIFO 0 watermark interrupt (MCAN_IR.RF0W).

>64: Watermark interrupt disabled.

- **F0OM: FIFO 0 Operation Mode**

FIFO 0 can be operated in Blocking or in Overwrite mode (see Section 50.5.4.2).

0: FIFO 0 Blocking mode.

1: FIFO 0 Overwrite mode.

50.6.28 MCAN Receive FIFO 0 Status

Name: MCAN_RXF0S

Address: 0xF80540A4 (0), 0xFC0500A4 (1)

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	RF0L	F0F
23	22	21	20	19	18	17	16
–	–	F0PI					
15	14	13	12	11	10	9	8
–	–	F0GI					
7	6	5	4	3	2	1	0
–	F0FL						

- **F0FL: Receive FIFO 0 Fill Level**

Number of elements stored in Receive FIFO 0, range 0 to 64.

- **F0GI: Receive FIFO 0 Get Index**

Receive FIFO 0 read index pointer, range 0 to 63.

- **F0PI: Receive FIFO 0 Put Index**

Receive FIFO 0 write index pointer, range 0 to 63.

- **F0F: Receive FIFO 0 Full**

0: Receive FIFO 0 not full.

1: Receive FIFO 0 full.

- **RF0L: Receive FIFO 0 Message Lost**

This bit is a copy of interrupt flag MCAN_IR.RF0L. When MCAN_IR.RF0L is reset, this bit is also reset.

0: No Receive FIFO 0 message lost

1: Receive FIFO 0 message lost, also set after write attempt to Receive FIFO 0 of size zero

Note: Overwriting the oldest message when MCAN_RXF0C.F0OM = ‘1’ will not set this flag.

50.6.29 MCAN Receive FIFO 0 Acknowledge

Name: MCAN_RXF0A

Address: 0xF80540A8 (0), 0xFC0500A8 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	F0AI					

- **F0AI: Receive FIFO 0 Acknowledge Index**

After the processor has read a message or a sequence of messages from Receive FIFO 0 it has to write the buffer index of the last element read from Receive FIFO 0 to F0AI. This will set the Receive FIFO 0 Get Index MCAN_RXF0S.F0GI to F0AI + 1 and update the FIFO 0 Fill Level MCAN_RXF0S.F0FL.

50.6.30 MCAN Receive Buffer Configuration

Name: MCAN_RXBC

Address: 0xF80540AC (0), 0xFC0500AC (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
RBSA							
7	6	5	4	3	2	1	0
RBSA						–	–

- **RBSA: Receive Buffer Start Address**

Configures the start address of the Receive Buffers section in the Message RAM (32-bit word address, see Figure 50-12). Also used to reference debug messages A,B,C.

Write RBSA with the bits [15:2] of the 32-bit address.

50.6.31 MCAN Receive FIFO 1 Configuration

Name: MCAN_RXF1C

Address: 0xF80540B0 (0), 0xFC0500B0 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
F1OM	F1WM						
23	22	21	20	19	18	17	16
–	F1S						
15	14	13	12	11	10	9	8
F1SA							
7	6	5	4	3	2	1	0
F1SA						–	–

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

- **F1SA: Receive FIFO 1 Start Address**

Start address of Receive FIFO 1 in Message RAM (32-bit word address, see Figure 50-12).

Write F1SA with the bits [15:2] of the 32-bit address.

- **F1S: Receive FIFO 1 Size**

0: No Receive FIFO 1

1-64: Number of elements in Receive FIFO 1.

>64: Values greater than 64 are interpreted as 64.

The elements in Receive FIFO 1 are indexed from 0 to F1S - 1.

- **F1WM: Receive FIFO 1 Watermark**

0: Watermark interrupt disabled

1-64: Level for Receive FIFO 1 watermark interrupt (MCAN_IR.RF1W).

>64: Watermark interrupt disabled.

- **F1OM: FIFO 1 Operation Mode**

FIFO 1 can be operated in Blocking or in Overwrite mode (see Section 50.5.4.2).

0: FIFO 1 Blocking mode.

1: FIFO 1 Overwrite mode.

50.6.32 MCAN Receive FIFO 1 Status

Name: MCAN_RXF1S

Address: 0xF80540B4 (0), 0xFC0500B4 (1)

Access: Read-only

31	30	29	28	27	26	25	24
DMS		–	–	–	–	RF1L	F1F
23	22	21	20	19	18	17	16
–	–	F1PI					
15	14	13	12	11	10	9	8
–	–	F1GI					
7	6	5	4	3	2	1	0
–	F1FL						

- **F1FL: Receive FIFO 1 Fill Level**

Number of elements stored in Receive FIFO 1, range 0 to 64.

- **F1GI: Receive FIFO 1 Get Index**

Receive FIFO 1 read index pointer, range 0 to 63.

- **F1PI: Receive FIFO 1 Put Index**

Receive FIFO 1 write index pointer, range 0 to 63.

- **F1F: Receive FIFO 1 Full**

0: Receive FIFO 1 not full.

1: Receive FIFO 1 full.

- **RF1L: Receive FIFO 1 Message Lost**

This bit is a copy of interrupt flag IR.RF1L. When IR.RF1L is reset, this bit is also reset.

0: No Receive FIFO 1 message lost.

1: Receive FIFO 1 message lost, also set after write attempt to Receive FIFO 1 of size zero.

Note: Overwriting the oldest message when MCAN_RXF1C.F1OM = ‘1’ will not set this flag.

- **DMS: Debug Message Status**

Value	Name	Description
0	IDLE	Idle state, wait for reception of debug messages, DMA request is cleared.
1	MSG_A	Debug message A received.
2	MSG_AB	Debug messages A, B received.
3	MSG_ABC	Debug messages A, B, C received, DMA request is set.

50.6.33 MCAN Receive FIFO 1 Acknowledge

Name: MCAN_RXF1A

Address: 0xF80540B8 (0), 0xFC0500B8 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	F1AI					

- **F1AI: Receive FIFO 1 Acknowledge Index**

After the processor has read a message or a sequence of messages from Receive FIFO 1 it has to write the buffer index of the last element read from Receive FIFO 1 to F1AI. This will set the Receive FIFO 1 Get Index MCAN_RXF1S.F1GI to F1AI + 1 and update the FIFO 1 Fill Level MCAN_RXF1S.F1FL.

50.6.34 MCAN Receive Buffer / FIFO Element Size Configuration

Name: MCAN_RXESC

Address: 0xF80540BC (0), 0xFC0500BC (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	RBDS		
7	6	5	4	3	2	1	0
–	F1DS			–	F0DS		

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

Configures the number of data bytes belonging to a Receive Buffer / Receive FIFO element. Data field sizes >8 bytes are intended for CAN FD operation only.

- **F0DS: Receive FIFO 0 Data Field Size**

Value	Name	Description
0	8_BYTE	8-byte data field
1	12_BYTE	12-byte data field
2	16_BYTE	16-byte data field
3	20_BYTE	20-byte data field
4	24_BYTE	24-byte data field
5	32_BYTE	32-byte data field
6	48_BYTE	48-byte data field
7	64_BYTE	64-byte data field

- **F1DS: Receive FIFO 1 Data Field Size**

Value	Name	Description
0	8_BYTE	8-byte data field
1	12_BYTE	12-byte data field
2	16_BYTE	16-byte data field
3	20_BYTE	20-byte data field
4	24_BYTE	24-byte data field
5	32_BYTE	32-byte data field
6	48_BYTE	48-byte data field
7	64_BYTE	64-byte data field

- **RBDS: Receive Buffer Data Field Size**

Value	Name	Description
0	8_BYTE	8-byte data field
1	12_BYTE	12-byte data field
2	16_BYTE	16-byte data field
3	20_BYTE	20-byte data field
4	24_BYTE	24-byte data field
5	32_BYTE	32-byte data field
6	48_BYTE	48-byte data field
7	64_BYTE	64-byte data field

Note: In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Receive Buffer or Receive FIFO, only the number of bytes as configured by MCAN_RXESC are stored to the Receive Buffer resp. Receive FIFO element. The rest of the frame's data field is ignored.

50.6.35 MCAN Tx Buffer Configuration

Name: MCAN_TXBC

Address: 0xF80540C0 (0), 0xFC0500C0 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	TFQM	TFQS					
23	22	21	20	19	18	17	16
–	–	NDTB					
15	14	13	12	11	10	9	8
TBSA							
7	6	5	4	3	2	1	0
TBSA						–	–

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

- **TBSA: Tx Buffers Start Address**

Start address of Tx Buffers section in Message RAM (32-bit word address, see Figure 50-12).

Write TBSA with the bits [15:2] of the 32-bit address.

- **NDTB: Number of Dedicated Transmit Buffers**

0: No dedicated Tx Buffers.

1-32: Number of dedicated Tx Buffers.

>32: Values greater than 32 are interpreted as 32.

- **TFQS: Transmit FIFO/Queue Size**

0: No Tx FIFO/Queue.

1-32: Number of Tx Buffers used for Tx FIFO/Queue.

>32: Values greater than 32 are interpreted as 32.

- **TFQM: Tx FIFO/Queue Mode**

0: Tx FIFO operation.

1: Tx Queue operation.

Note: The sum of TFQS and NDTB may be not greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.

50.6.36 MCAN Tx FIFO/Queue Status

Name: MCAN_TXFQS

Address: 0xF80540C4 (0), 0xFC0500C4 (1)

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	TFQF	TFQPI				
15	14	13	12	11	10	9	8
–	–	–	TFGI				
7	6	5	4	3	2	1	0
–	–	TFFL					

The Tx FIFO/Queue status is related to the pending Tx requests listed in register MCAN_TXBRP. Therefore the effect of Add/Cancellation requests may be delayed due to a running Tx scan (MCAN_TXBRP not yet updated).

- **TFFL: Tx FIFO Free Level**

Number of consecutive free Tx FIFO elements starting from TFGI, range 0 to 32. Read as zero when Tx Queue operation is configured (MCAN_TXBC.TFQM = ‘1’).

- **TFGI: Tx FIFO Get Index**

Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (MCAN_TXBC.TFQM = ‘1’).

- **TFQPI: Tx FIFO/Queue Put Index**

Tx FIFO/Queue write index pointer, range 0 to 31.

- **TFQF: Tx FIFO/Queue Full**

0: Tx FIFO/Queue not full.

1: Tx FIFO/Queue full.

Note: In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers.
Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.

50.6.37 MCAN Tx Buffer Element Size Configuration

Name: MCAN_TXESC

Address: 0xF80540C8 (0), 0xFC0500C8 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	TBDS		

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes > 8 bytes are intended for CAN FD operation only.

- **TBDS: Tx Buffer Data Field Size**

Value	Name	Description
0	8_BYTE	8-byte data field
1	12_BYTE	12-byte data field
2	16_BYTE	16-byte data field
3	20_BYTE	20-byte data field
4	24_BYTE	24-byte data field
5	32_BYTE	32-byte data field
6	48_BYTE	48- byte data field
7	64_BYTE	64-byte data field

Note: In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size MCAN_TXESC.TBDS, the bytes not defined by the Tx Buffer are transmitted as “0xCC” (padding bytes).

50.6.38 MCAN Transmit Buffer Request Pending

Name: MCAN_TXBRP

Address: 0xF80540CC (0), 0xFC0500CC (1)

Access: Read-only

31	30	29	28	27	26	25	24
TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
23	22	21	20	19	18	17	16
TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
15	14	13	12	11	10	9	8
TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
7	6	5	4	3	2	1	0
TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0

- **TRPx: Transmission Request Pending for Buffer x**

Each Tx Buffer has its own Transmission Request Pending bit. The bits are set via register MCAN_TXBAR. The bits are reset after a requested transmission has completed or has been cancelled via register MCAN_TXBCR.

TXBRP bits are set only for those Tx Buffers configured via MCAN_TXBC. After a MCAN_TXBRP bit has been set, a Tx scan (see Section 50.5.5) is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID).

A cancellation request resets the corresponding transmission request pending bit of register MCAN_TXBRP. In case a transmission has already been started when a cancellation is requested, this is done at the end of the transmission, regardless whether the transmission was successful or not. The cancellation request bits are reset directly after the corresponding TXBRP bit has been reset.

After a cancellation has been requested, a finished cancellation is signalled via MCAN_TXBCF.

- after successful transmission together with the corresponding MCAN_TXBTO bit.
- when the transmission has not yet been started at the point of cancellation.
- when the transmission has been aborted due to lost arbitration.
- when an error occurred during frame transmission.

In DAR mode, all transmissions are automatically cancelled if they are not successful. The corresponding MCAN_TXBCF bit is set for all unsuccessful transmissions.

0: No transmission request pending

1: Transmission request pending

Note: MCAN_TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is cancelled immediately, the corresponding MCAN_TXBRP bit is reset.

50.6.39 MCAN Transmit Buffer Add Request

Name: MCAN_TXBAR

Address: 0xF80540D0 (0), 0xFC0500D0 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
23	22	21	20	19	18	17	16
AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
15	14	13	12	11	10	9	8
AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
7	6	5	4	3	2	1	0
AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0

- **ARx: Add Request for Transmit Buffer x**

Each Transmit Buffer has its own Add Request bit. Writing a ‘1’ will set the corresponding Add Request bit; writing a ‘0’ has no impact. This enables the processor to set transmission requests for multiple Transmit Buffers with one write to MCAN_TXBAR. MCAN_TXBAR bits are set only for those Transmit Buffers configured via TXBC. When no Transmit scan is running, the bits are reset immediately, else the bits remain set until the Transmit scan process has completed.

0: No transmission request added.

1: Transmission requested added.

Note: If an add request is applied for a Transmit Buffer with pending transmission request (corresponding MCAN_TXBRP bit already set), this Add Request is ignored.

50.6.40 MCAN Transmit Buffer Cancellation Request

Name: MCAN_TXBCR

Address: 0xF80540D4 (0), 0xFC0500D4 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
23	22	21	20	19	18	17	16
CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
15	14	13	12	11	10	9	8
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
7	6	5	4	3	2	1	0
CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

- **CRx: Cancellation Request for Transmit Buffer x**

Each Transmit Buffer has its own Cancellation Request bit. Writing a ‘1’ will set the corresponding Cancellation Request bit; writing a ‘0’ has no impact. This enables the processor to set cancellation requests for multiple Transmit Buffers with one write to MCAN_TXBCR. MCAN_TXBCR bits are set only for those Transmit Buffers configured via TXBC. The bits remain set until the corresponding bit of MCAN_TXBRP is reset.

0: No cancellation pending.

1: Cancellation pending.

50.6.41 MCAN Transmit Buffer Transmission Occurred

Name: MCAN_TXBTO

Address: 0xF80540D8 (0), 0xFC0500D8 (1)

Access: Read-only

31	30	29	28	27	26	25	24
TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
23	22	21	20	19	18	17	16
TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
15	14	13	12	11	10	9	8
TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
7	6	5	4	3	2	1	0
TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0

- **TOx: Transmission Occurred for Buffer x**

Each Transmit Buffer has its own Transmission Occurred bit. The bits are set when the corresponding MCAN_TXBRP bit is cleared after a successful transmission. The bits are reset when a new transmission is requested by writing a ‘1’ to the corresponding bit of register MCAN_TXBAR.

0: No transmission occurred.

1: Transmission occurred.

50.6.42 MCAN Transmit Buffer Cancellation Finished

Name: MCAN_TXBCF

Address: 0xF80540DC (0), 0xFC0500DC (1)

Access: Read-only

31	30	29	28	27	26	25	24
CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
23	22	21	20	19	18	17	16
CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
15	14	13	12	11	10	9	8
CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
7	6	5	4	3	2	1	0
CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0

- **CFx: Cancellation Finished for Transmit Buffer x**

Each Transmit Buffer has its own Cancellation Finished bit. The bits are set when the corresponding MCAN_TXBRP bit is cleared after a cancellation was requested via MCAN_TXBCR. In case the corresponding MCAN_TXBRP bit was not set at the point of cancellation, CF is set immediately. The bits are reset when a new transmission is requested by writing a ‘1’ to the corresponding bit of register MCAN_TXBAR.

0: No transmit buffer cancellation.

1: Transmit buffer cancellation finished.

50.6.43 MCAN Transmit Buffer Transmission Interrupt Enable

Name: MCAN_TXBTIE

Address: 0xF80540E0 (0), 0xFC0500E0 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
23	22	21	20	19	18	17	16
TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
15	14	13	12	11	10	9	8
TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
7	6	5	4	3	2	1	0
TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0

- **TIEx: Transmission Interrupt Enable for Buffer x**

Each Transmit Buffer has its own Transmission Interrupt Enable bit.

0: Transmission interrupt disabled

1: Transmission interrupt enable

50.6.44 MCAN Transmit Buffer Cancellation Finished Interrupt Enable

Name: MCAN_TXBCIE

Address: 0xF80540E4 (0), 0xFC0500E4 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
23	22	21	20	19	18	17	16
CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
15	14	13	12	11	10	9	8
CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
7	6	5	4	3	2	1	0
CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0

- **CFIEx: Cancellation Finished Interrupt Enable for Transmit Buffer x**

Each Transmit Buffer has its own Cancellation Finished Interrupt Enable bit.

0: Cancellation finished interrupt disabled.

1: Cancellation finished interrupt enabled.

50.6.45 MCAN Transmit Event FIFO Configuration

Name: MCAN_TXEFC

Address: 0xF80540F0 (0), 0xFC0500F0 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	EFWM					
23	22	21	20	19	18	17	16
–	–	EFS					
15	14	13	12	11	10	9	8
EFSA							
7	6	5	4	3	2	1	0
EFSA						–	–

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

- **EFSA: Event FIFO Start Address**

Start address of Tx Event FIFO in Message RAM (32-bit word address, see Figure 50-12).

Write EFSA with the bits [15:2] of the 32-bit address.

- **EFS: Event FIFO Size**

0: Tx Event FIFO disabled.

1-32: Number of Tx Event FIFO elements.

>32: Values greater than 32 are interpreted as 32.

The Tx Event FIFO elements are indexed from 0 to EFS - 1.

- **EFWM: Event FIFO Watermark**

0: Watermark interrupt disabled.

1-32: Level for Tx Event FIFO watermark interrupt (MCAN_IR.TEFW).

>32: Watermark interrupt disabled.

50.6.46 MCAN Tx Event FIFO Status

Name: MCAN_TXEFS

Address: 0xF80540F4 (0), 0xFC0500F4 (1)

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	TEFL	EFF
23	22	21	20	19	18	17	16
–	–	–	EFPI				
15	14	13	12	11	10	9	8
–	–	–	EFGI				
7	6	5	4	3	2	1	0
–	–	EFFL					

- **EFFL: Event FIFO Fill Level**

Number of elements stored in Tx Event FIFO, range 0 to 32.

- **EFGI: Event FIFO Get Index**

Tx Event FIFO read index pointer, range 0 to 31.

- **EFPI: Event FIFO Put Index**

Tx Event FIFO write index pointer, range 0 to 31.

- **EFF: Event FIFO Full**

0: Tx Event FIFO not full

1: Tx Event FIFO full

- **TEFL: Tx Event FIFO Element Lost**

This bit is a copy of interrupt flag MCAN_IR.TEFL. When MCAN_IR.TEFL is reset, this bit is also reset.

0: No Tx Event FIFO element lost

1: Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.

50.6.47 MCAN Tx Event FIFO Acknowledge

Name: MCAN_TXEFA

Address: 0xF80540F8 (0), 0xFC0500F8 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	EFAI				

- **EFAI: Event FIFO Acknowledge Index**

After the processor has read an element or a sequence of elements from the Tx Event FIFO, it has to write the index of the last element read from Tx Event FIFO to EFAI. This will set the Tx Event FIFO Get Index MCAN_TXEFS.EFGI to EFAI + 1 and update the FIFO 0 Fill Level MCAN_TXEFS.EFFL.

51. Timer Counter (TC)

51.1 Description

A Timer Counter (TC) module includes three identical TC channels. The number of implemented TC modules is device-specific.

Each TC channel can be independently programmed to perform a wide range of functions including frequency measurement, event counting, interval measurement, pulse generation, delay timing and pulse width modulation.

Each channel has three external clock inputs, five internal clock inputs and two multipurpose input/output signals which can be configured by the user. Each channel drives an internal interrupt signal which can be programmed to generate processor interrupts.

The TC embeds a quadrature decoder (QDEC) connected in front of the timers and driven by TIOA0, TIOB0 and TIOB1 inputs. When enabled, the QDEC performs the input lines filtering, decoding of quadrature signals and connects to the timers/counters in order to read the position and speed of the motor through the user interface.

The TC block has two global registers which act upon all TC channels:

- Block Control Register (TC_BCR)—allows channels to be started simultaneously with the same instruction
- Block Mode Register (TC_BMR)—defines the external clock inputs for each channel, allowing them to be chained

51.2 Embedded Characteristics

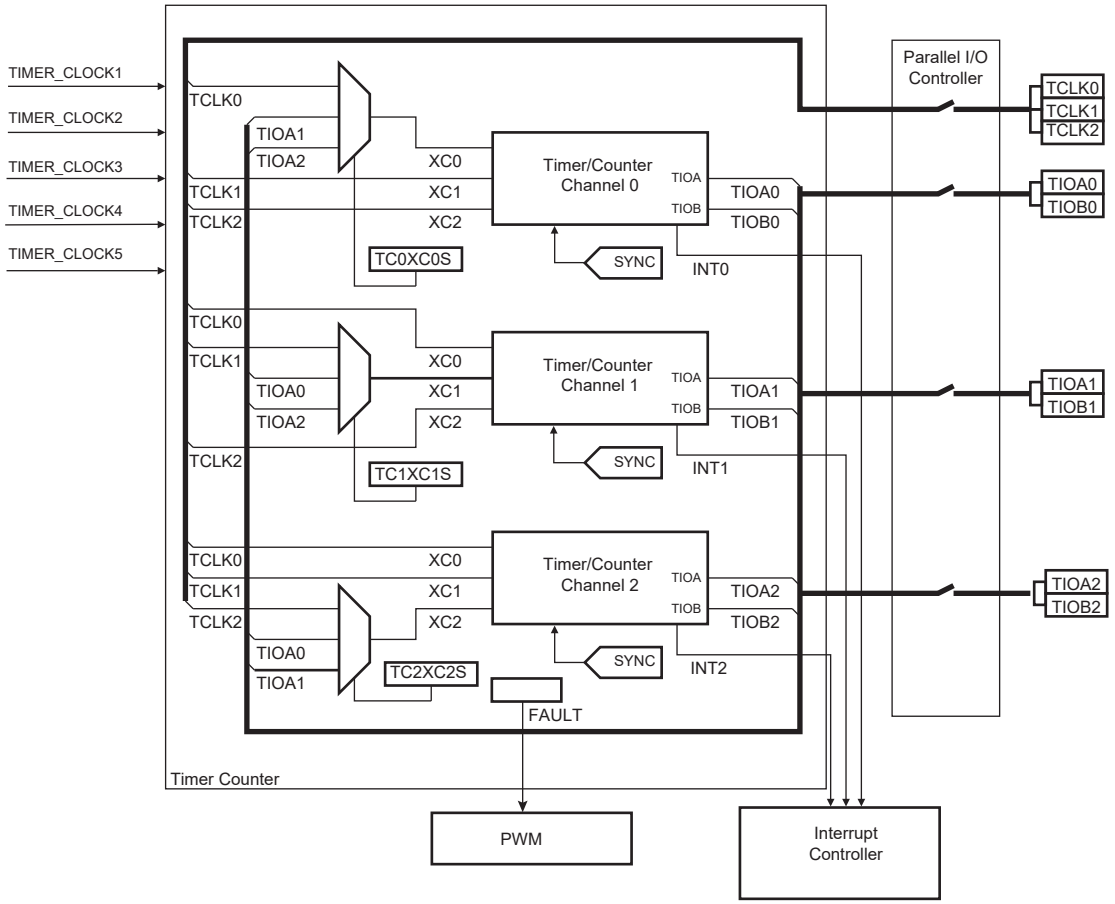
- Total number of TC channels implemented on this device: 6
- TC channel size: 32-bit
- Wide range of functions including:
  - Frequency measurement
  - Event counting
  - Interval measurement
  - Pulse generation
  - Delay timing
  - Pulse Width Modulation
  - Up/down capabilities
  - Quadrature decoder
  - 2-bit Gray up/down count for stepper motor
- Each channel is user-configurable and contains:
  - Three external clock inputs
  - Five Internal clock inputs
  - Two multipurpose input/output signals acting as trigger event
  - Trigger/capture events can be directly synchronized by PWM signals
- Internal interrupt signal
- Read of the Capture registers by the DMAC
- Compare event fault generation for PWM
- Register Write Protection

51.3 Block Diagram

Table 51-1. Timer Counter Clock Assignment

Name	Definition
TIMER_CLOCK1	GCLK [35], GCLK [36]
TIMER_CLOCK2	System bus clock divided by 8
TIMER_CLOCK3	System bus clock divided by 32
TIMER_CLOCK4	System bus clock divided by 128
TIMER_CLOCK5	slow_clock

Figure 51-1. Timer Counter Block Diagram



Note: The QDEC connections are detailed in Figure 51-17.

Table 51-2. Channel Signal Description

Signal Name	Description
XC0, XC1, XC2	External Clock Inputs
TIOAx	Capture Mode: Timer Counter Input Waveform Mode: Timer Counter Output
TIOBx	Capture Mode: Timer Counter Input Waveform Mode: Timer Counter Input/Output

Table 51-2. Channel Signal Description (Continued)

Signal Name	Description
INT	Interrupt Signal Output (internal signal)
SYNC	Synchronization Input Signal (from configuration register)

51.4 Pin List

Table 51-3. Pin List

Pin Name	Description	Type
TCLK0–TCLK2	External Clock Input	Input
TIOA0–TIOA2	I/O Line A	I/O
TIOB0–TIOB2	I/O Line B	I/O

51.5 Product Dependencies

51.5.1 I/O Lines

The pins used for interfacing the compliant external devices may be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the TC pins to their peripheral functions.

Table 51-4. I/O Lines

Instance	Signal	I/O Line	Peripheral
TC0	TCLK0	PA21	D
TC0	TCLK1	PA29	A
TC0	TCLK1	PC5	C
TC0	TCLK1	PD13	A
TC0	TCLK2	PB5	A
TC0	TCLK2	PB24	D
TC0	TCLK2	PD22	A
TC0	TIOA0	PA19	D
TC0	TIOA1	PA27	A
TC0	TIOA1	PC3	C
TC0	TIOA1	PD11	A
TC0	TIOA2	PB6	A
TC0	TIOA2	PB22	D
TC0	TIOA2	PD20	A
TC0	TIOB0	PA20	D
TC0	TIOB1	PA28	A
TC0	TIOB1	PC4	C
TC0	TIOB1	PD12	A
TC0	TIOB2	PB7	A
TC0	TIOB2	PB23	D

Table 51-4. I/O Lines (Continued)

Instance	Signal	I/O Line	Peripheral
TC0	TIOB2	PD21	A
TC1	TCLK3	PB8	A
TC1	TCLK3	PB21	D
TC1	TCLK3	PD31	D
TC1	TCLK4	PA11	D
TC1	TCLK4	PC11	D
TC1	TCLK5	PA8	D
TC1	TCLK5	PB30	D
TC1	TIOA3	PB9	A
TC1	TIOA3	PB19	D
TC1	TIOA3	PD29	D
TC1	TIOA4	PA9	D
TC1	TIOA4	PC9	D
TC1	TIOA5	PA6	D
TC1	TIOA5	PB28	D
TC1	TIOB3	PB10	A
TC1	TIOB3	PB20	D
TC1	TIOB3	PD30	D
TC1	TIOB4	PA10	D
TC1	TIOB4	PC10	D
TC1	TIOB5	PA7	D
TC1	TIOB5	PB29	D

51.5.2 Power Management

The TC is clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the Timer Counter clock.

51.5.3 Interrupt Sources

The TC has an interrupt line connected to the interrupt controller. Handling the TC interrupt requires programming the interrupt controller before configuring the TC.

Table 51-5. Peripheral IDs

Instance	ID
TC0	35
TC1	36

51.5.4 Synchronization Inputs from PWM

The TC has trigger/capture inputs internally connected to the PWM. Refer to Section 51.6.14 “Synchronization with PWM” and to the implementation of the Pulse Width Modulation (PWM) in this product.

51.5.5 Fault Output

The TC has the FAULT output internally connected to the fault input of PWM. Refer to Section 51.6.18 “Fault Mode” and to the implementation of the Pulse Width Modulation (PWM) in this product.

51.6 Functional Description

51.6.1 Description

All channels of the Timer Counter are independent and identical in operation except when the QDEC is enabled. The registers for channel programming are listed in Table 51-6 “Register Mapping”.

51.6.2 32-bit Counter

Each 32-bit channel is organized around a 32-bit counter. The value of the counter is incremented at each positive edge of the selected clock. When the counter has reached the value 2^{32}-1 and passes to zero, an overflow occurs and the COVFS bit in the TC Status Register (TC_SR) is set.

The current value of the counter is accessible in real time by reading the TC Counter Value Register (TC_CV). The counter can be reset by a trigger. In this case, the counter value passes to zero on the next valid edge of the selected clock.

51.6.3 Clock Selection

At block level, input clock signals of each channel can be connected either to the external inputs TCLKx, or to the internal I/O signals TIOAx for chaining(1) by programming the TC Block Mode Register (TC_BMR). See Figure 51-2.

Each channel can independently select an internal or external clock source for its counter(2):

- External clock signals: XC0, XC1 or XC2
- Internal clock signals: GCLK [35], GCLK [36], System bus clock divided by 8, System bus clock divided by 32, System bus clock divided by 128, slow_clock

This selection is made by the TCCLKS bits in the TC Channel Mode Register (TC_CMR).

The selected clock can be inverted with the CLKI bit in the TC_CMR. This allows counting on the opposite edges of the clock.

The burst function allows the clock to be validated when an external signal is high. The BURST parameter in the TC_CMR defines this signal (none, XC0, XC1, XC2). See Figure 51-3.

Notes:
1. In Waveform mode, to chain two timers, it is mandatory to initialize some parameters:
   - Configure TIOx outputs to 1 or 0 by writing the required value to TC_CMR.ASWTRG.
   - Bit TC_BCR.SYNC must be written to 1 to start the channels at the same time.
2. In all cases, if an external clock or asynchronous internal clock GCLK is used, the duration of each of its levels must be longer than the peripheral clock period, so the clock frequency will be at least 2.5 times lower than the peripheral clock.

Figure 51-2. Clock Chaining Selection

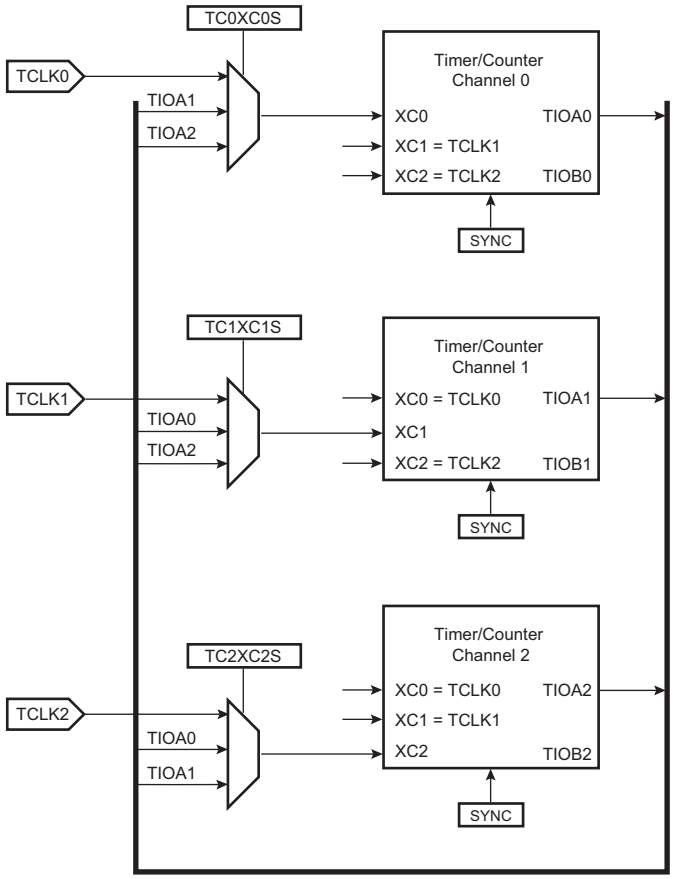
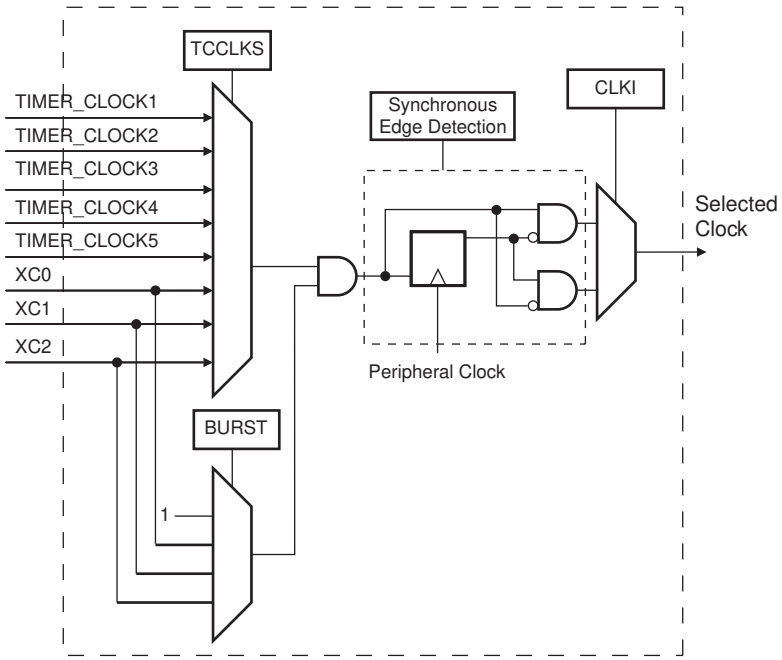


Figure 51-3. Clock Selection

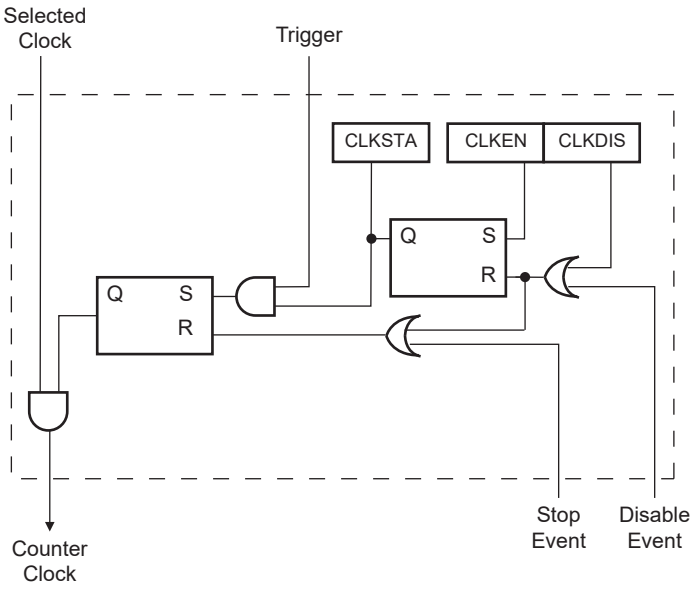


51.6.4 Clock Control

The clock of each counter can be controlled in two different ways: it can be enabled/disabled and started/stopped. See Figure 51-4.

- The clock can be enabled or disabled by the user with the CLKEN and the CLKDIS commands in the TC Channel Control Register (TC_CCR). In Capture mode it can be disabled by an RB load event if LDBDIS is set to 1 in the TC_CMR. In Waveform mode, it can be disabled by an RC Compare event if CPCDIS is set to 1 in TC_CMR. When disabled, the start or the stop actions have no effect: only a CLKEN command in the TC_CCR can re-enable the clock. When the clock is enabled, the CLKSTA bit is set in the TC_SR.
- The clock can also be started or stopped: a trigger (software, synchro, external or compare) always starts the clock. The clock can be stopped by an RB load event in Capture mode (LDBSTOP = 1 in TC_CMR) or an RC compare event in Waveform mode (CPCSTOP = 1 in TC_CMR). The start and the stop commands are effective only if the clock is enabled.

Figure 51-4. Clock Control



51.6.5 Operating Modes

Each channel can operate independently in two different modes:

- Capture mode provides measurement on signals.
- Waveform mode provides wave generation.

The TC operating mode is programmed with the WAVE bit in the TC_CMR.

In Capture mode, TIOAx and TIOBx are configured as inputs.

In Waveform mode, TIOAx is always configured to be an output and TIOBx is an output if it is not selected to be the external trigger.

51.6.6 Trigger

A trigger resets the counter and starts the counter clock. Three types of triggers are common to both modes, and a fourth external trigger is available to each mode.

Regardless of the trigger used, it will be taken into account at the following active edge of the selected clock. This means that the counter value can be read differently from zero just after a trigger, especially when a low frequency signal is selected as the clock.

The following triggers are common to both modes:

- Software Trigger: Each channel has a software trigger, available by setting SWTRG in TC_CCR.
- SYNC: Each channel has a synchronization signal SYNC. When asserted, this signal has the same effect as a software trigger. The SYNC signals of all channels are asserted simultaneously by writing TC_BCR (Block Control) with SYNC set.
- Compare RC Trigger: RC is implemented in each channel and can provide a trigger when the counter value matches the RC value if CPCTRG is set in the TC_CMR.

The channel can also be configured to have an external trigger. In Capture mode, the external trigger signal can be selected between TIOAx and TIOBx. In Waveform mode, an external event can be programmed on one of the following signals: TIOBx, XC0, XC1 or XC2. This external event can then be programmed to perform a trigger by setting bit ENETRG in the TC_CMR.

If an external trigger is used, the duration of the pulses must be longer than the peripheral clock period in order to be detected.

51.6.7 Capture Mode

Capture mode is entered by clearing the WAVE bit in the TC_CMR.

Capture mode allows the TC channel to perform measurements such as pulse timing, frequency, period, duty cycle and phase on TIOAx and TIOBx signals which are considered as inputs.

Figure 51-6 shows the configuration of the TC channel when programmed in Capture mode.

51.6.8 Capture Registers A and B

Registers A and B (RA and RB) are used as capture registers. They can be loaded with the counter value when a programmable event occurs on the signal TIOAx.

The LDRA field in the TC_CMR defines the TIOAx selected edge for the loading of register A, and the LDRB field defines the TIOAx selected edge for the loading of Register B.

The subsampling ratio defined by the SBSMPLR field in TC_CMR is applied to these selected edges, so that the loading of Register A and Register B occurs once every 1, 2, 4, 8 or 16 selected edges.

RA is loaded only if it has not been loaded since the last trigger or if RB has been loaded since the last loading of RA.

RB is loaded only if RA has been loaded since the last trigger or the last loading of RB.

Loading RA or RB before the read of the last value loaded sets the Overrun Error Flag (LOVRS bit) in the TC_SR. In this case, the old value is overwritten.

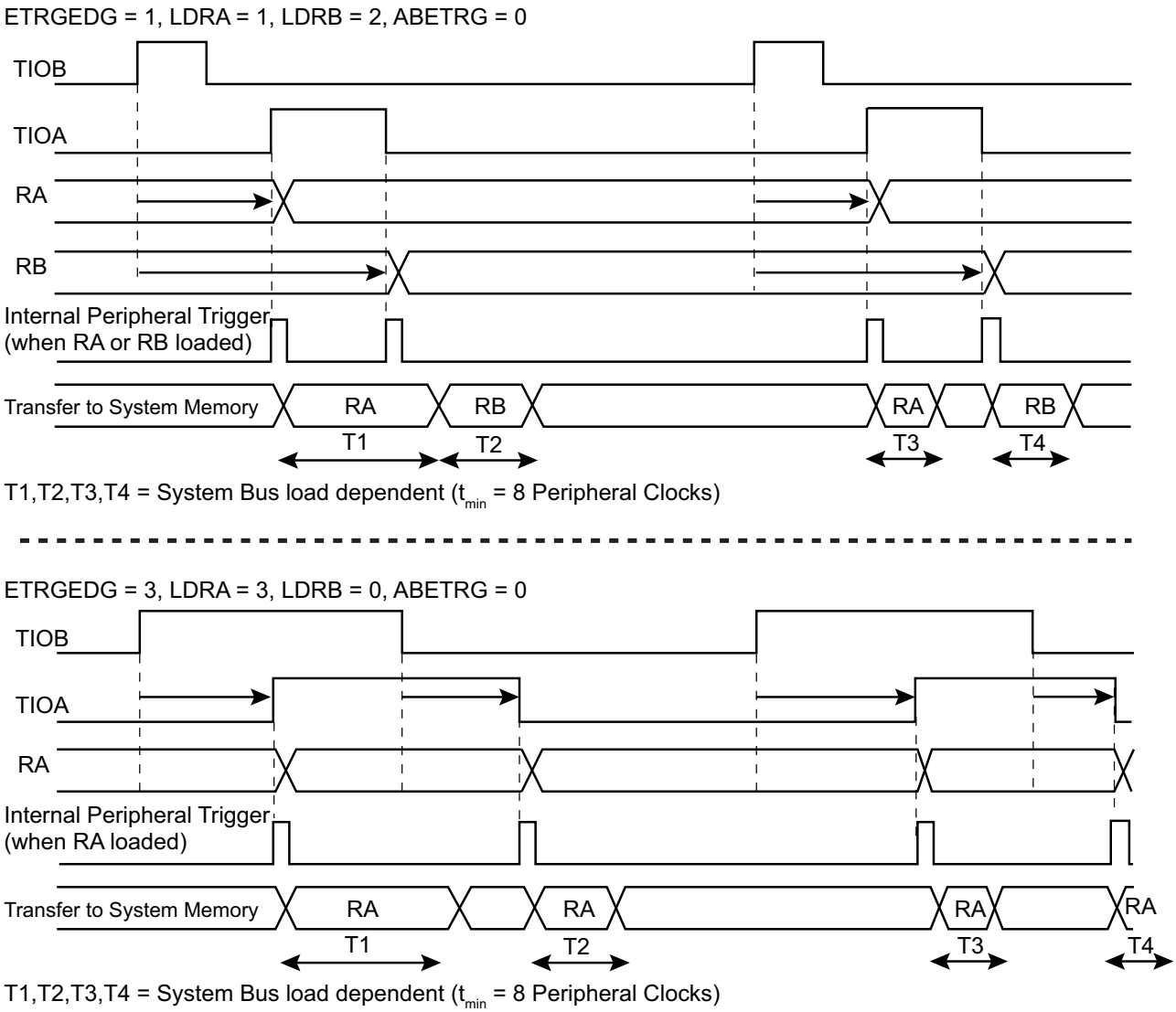
When DMA is used, the RAB register address must be configured as source address of the transfer. The RAB register provides the next unread value from Register A and Register B. It may be read by the DMA after a request has been triggered upon loading Register A or Register B.

51.6.9 Transfer with DMAC in Capture Mode

The DMAC can perform access from the TC to system memory in Capture mode only.

Figure 51-5 illustrates how TC_RA and TC_RB can be loaded in the system memory without CPU intervention.

Figure 51-5. Example of Transfer with DMAC in Capture Mode

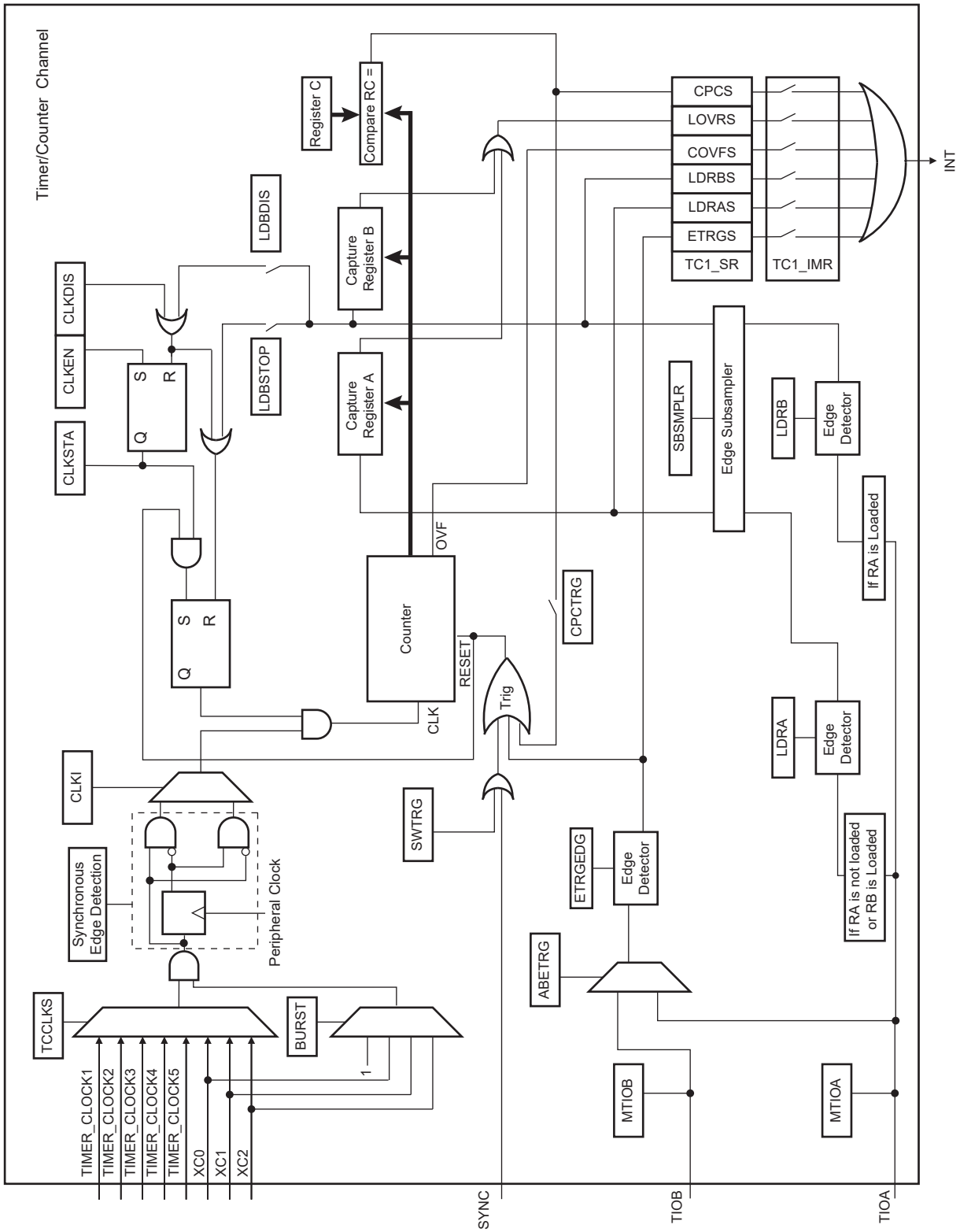


51.6.10 Trigger Conditions

In addition to the SYNC signal, the software trigger and the RC compare trigger, an external trigger can be defined.

The ABETRG bit in the TC_CMR selects TIOAx or TIOBx input signal as an external trigger or the trigger signal from the output comparator of the PWM module. The External Trigger Edge Selection parameter (ETRGEDG field in TC_CMR) defines the edge (rising, falling, or both) detected to generate an external trigger. If ETRGEDG = 0 (none), the external trigger is disabled.

Figure 51-6. Capture Mode



51.6.11 Waveform Mode

Waveform mode is entered by setting the TC_CMRx.WAVE bit.

In Waveform mode, the TC channel generates one or two PWM signals with the same frequency and independently programmable duty cycles, or generates different types of one-shot or repetitive pulses.

In this mode, TIOAx is configured as an output and TIOBx is defined as an output if it is not used as an external event (EEVT parameter in TC_CMR).

Figure 51-7 shows the configuration of the TC channel when programmed in Waveform operating mode.

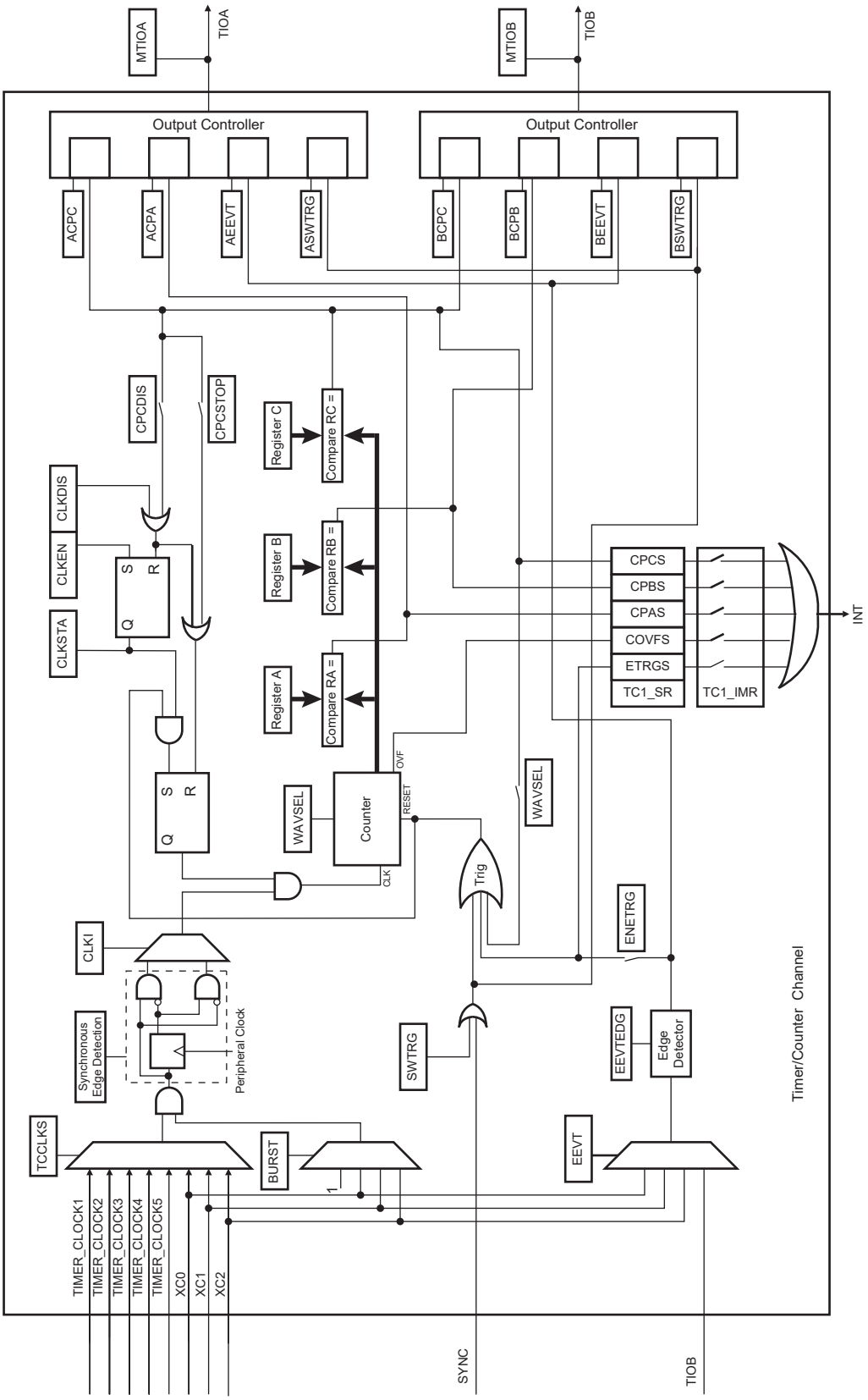
51.6.12 Waveform Selection

Depending on the WAVSEL parameter in TC_CMR, the behavior of TC_CV varies.

With any selection, TC_RA, TC_RB and TC_RC can all be used as compare registers.

RA Compare is used to control the TIOAx output, RB Compare is used to control the TIOBx output (if correctly configured) and RC Compare is used to control TIOAx and/or TIOBx outputs.

Figure 51-7. Waveform Mode



51.6.12.1 WAVSEL = 00

When WAVSEL = 00, the value of TC_CV is incremented from 0 to 2^{32}-1. Once 2^{32}-1 has been reached, the value of TC_CV is reset. Incrementation of TC_CV starts again and the cycle continues. See Figure 51-8.

An external event trigger or a software trigger can reset the value of TC_CV. It is important to note that the trigger may occur at any time. See Figure 51-9.

RC Compare cannot be programmed to generate a trigger in this configuration. At the same time, RC Compare can stop the counter clock (CPCSTOP = 1 in TC_CMR) and/or disable the counter clock (CPCDIS = 1 in TC_CMR).

Figure 51-8. WAVSEL = 00 without Trigger

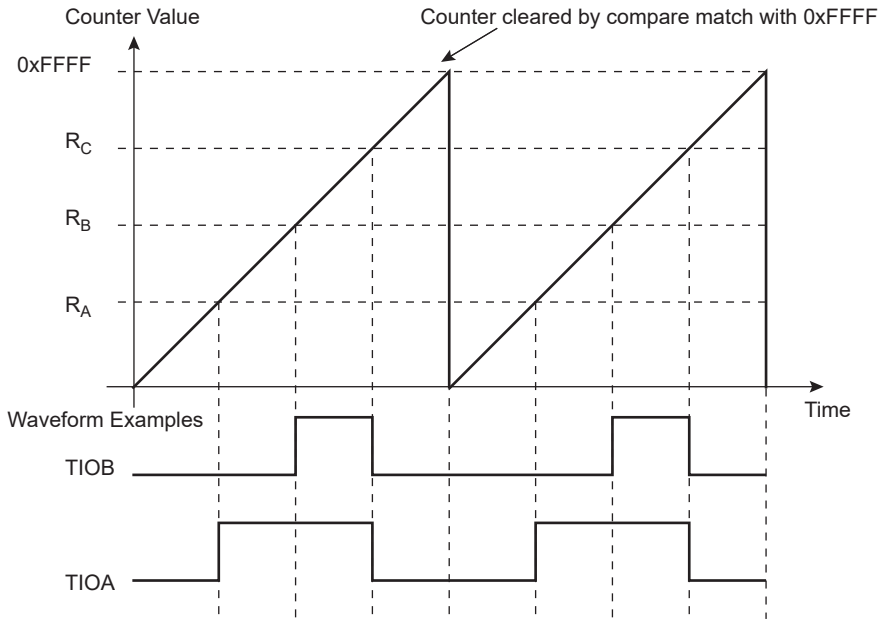
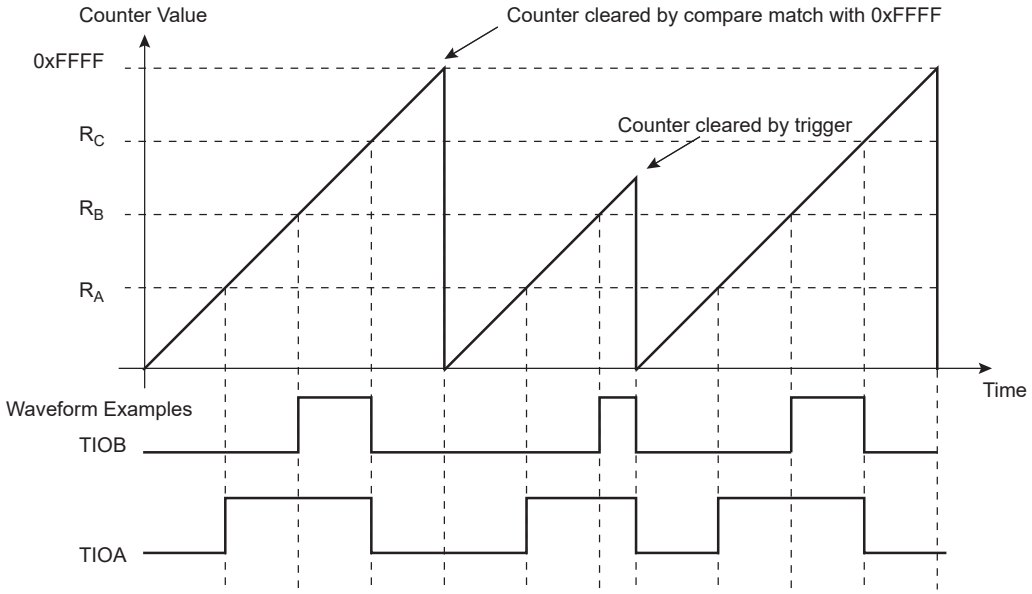


Figure 51-9. WAVSEL = 00 with Trigger



51.6.12.2 WAVSEL = 10

When WAVSEL = 10, the value of TC_CV is incremented from 0 to the value of RC, then automatically reset on a RC Compare. Once the value of TC_CV has been reset, it is then incremented and so on. See Figure 51-10.

It is important to note that TC_CV can be reset at any time by an external event or a software trigger if both are programmed correctly. See Figure 51-11.

In addition, RC Compare can stop the counter clock (CPCSTOP = 1 in TC_CMR) and/or disable the counter clock (CPCDIS = 1 in TC_CMR).

Figure 51-10. WAVSEL = 10 without Trigger

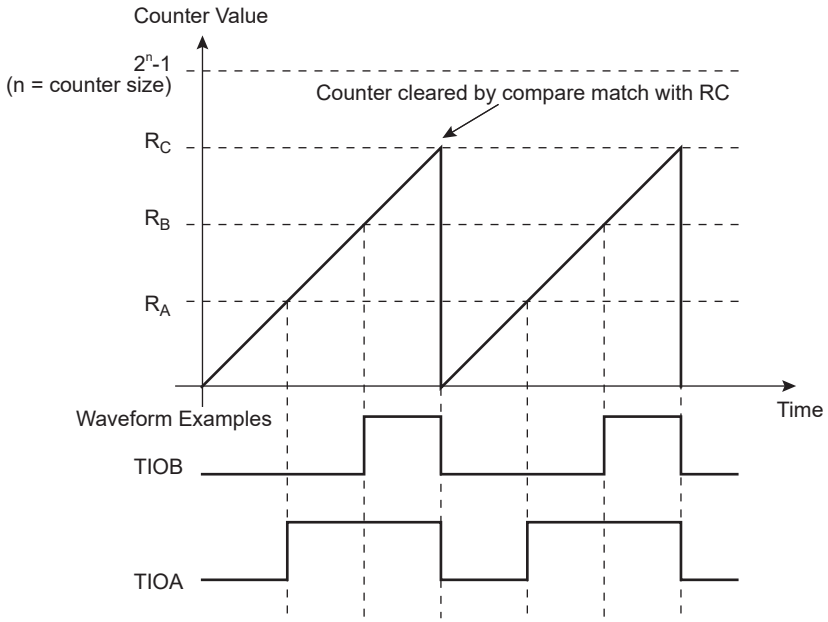
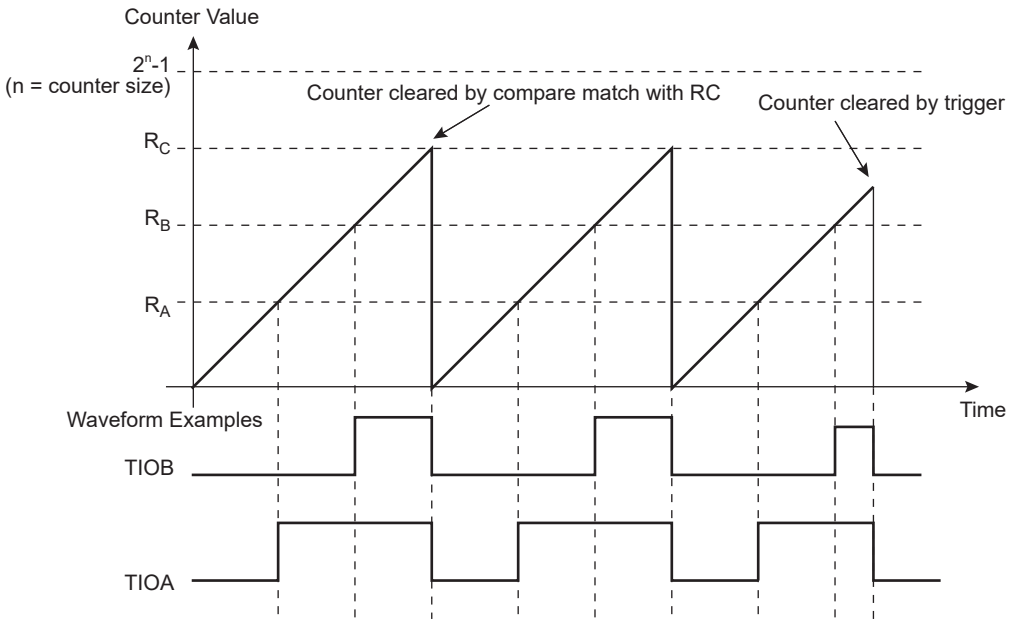


Figure 51-11. WAVSEL = 10 with Trigger



51.6.12.3 WAVSEL = 01

When WAVSEL = 01, the value of TC_CV is incremented from 0 to 2^{32}-1 . Once 2^{32}-1 is reached, the value of TC_CV is decremented to 0, then reincremented to 2^{32}-1 and so on. See Figure 51-12.

A trigger such as an external event or a software trigger can modify TC_CV at any time. If a trigger occurs while TC_CV is incrementing, TC_CV then decrements. If a trigger is received while TC_CV is decrementing, TC_CV then increments. See Figure 51-13.

RC Compare cannot be programmed to generate a trigger in this configuration.

At the same time, RC Compare can stop the counter clock (CPCSTOP = 1) and/or disable the counter clock (CPCDIS = 1).

Figure 51-12. WAVSEL = 01 without Trigger

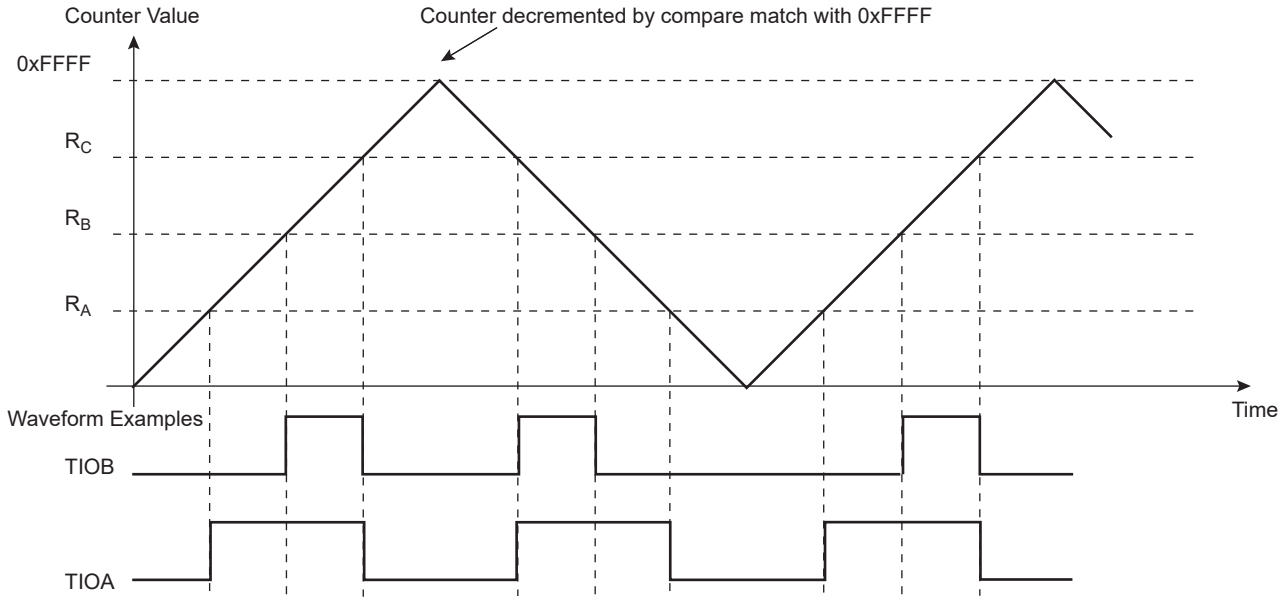
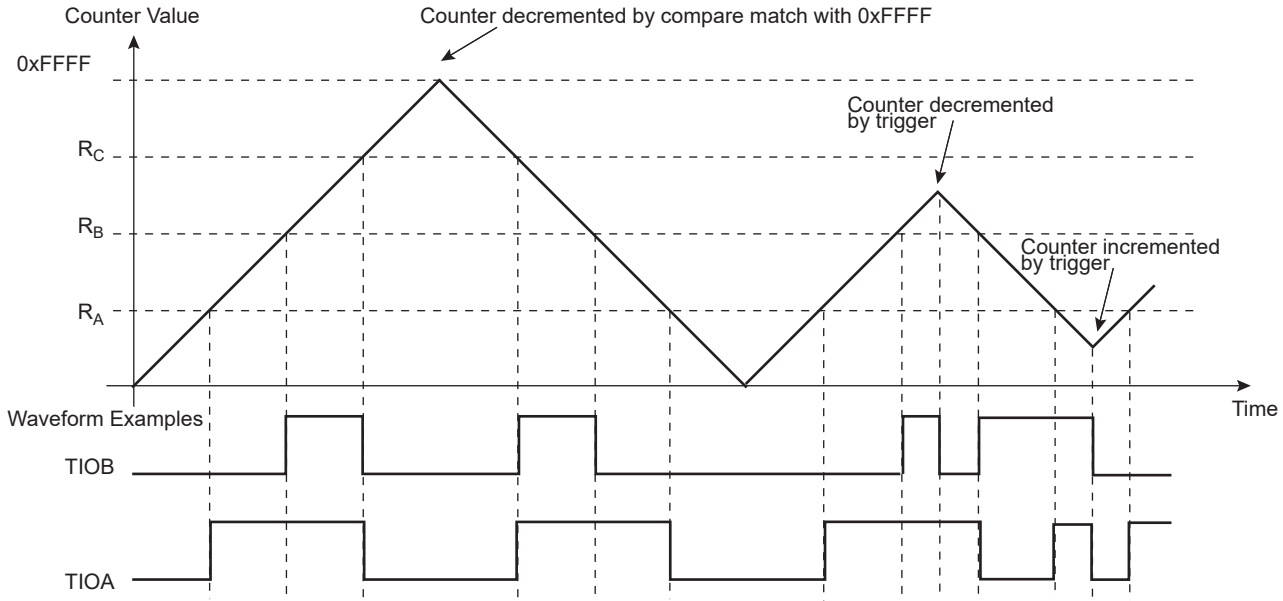


Figure 51-13. WAVSEL = 01 with Trigger



51.6.12.4 WAVSEL = 11

When WAVSEL = 11, the value of TC_CV is incremented from 0 to RC. Once RC is reached, the value of TC_CV is decremented to 0, then reincremented to RC and so on. See Figure 51-14.

A trigger such as an external event or a software trigger can modify TC_CV at any time. If a trigger occurs while TC_CV is incrementing, TC_CV then decrements. If a trigger is received while TC_CV is decrementing, TC_CV then increments. See Figure 51-15.

RC Compare can stop the counter clock (CPCSTOP = 1) and/or disable the counter clock (CPCDIS = 1).

Figure 51-14. WAVSEL = 11 without Trigger

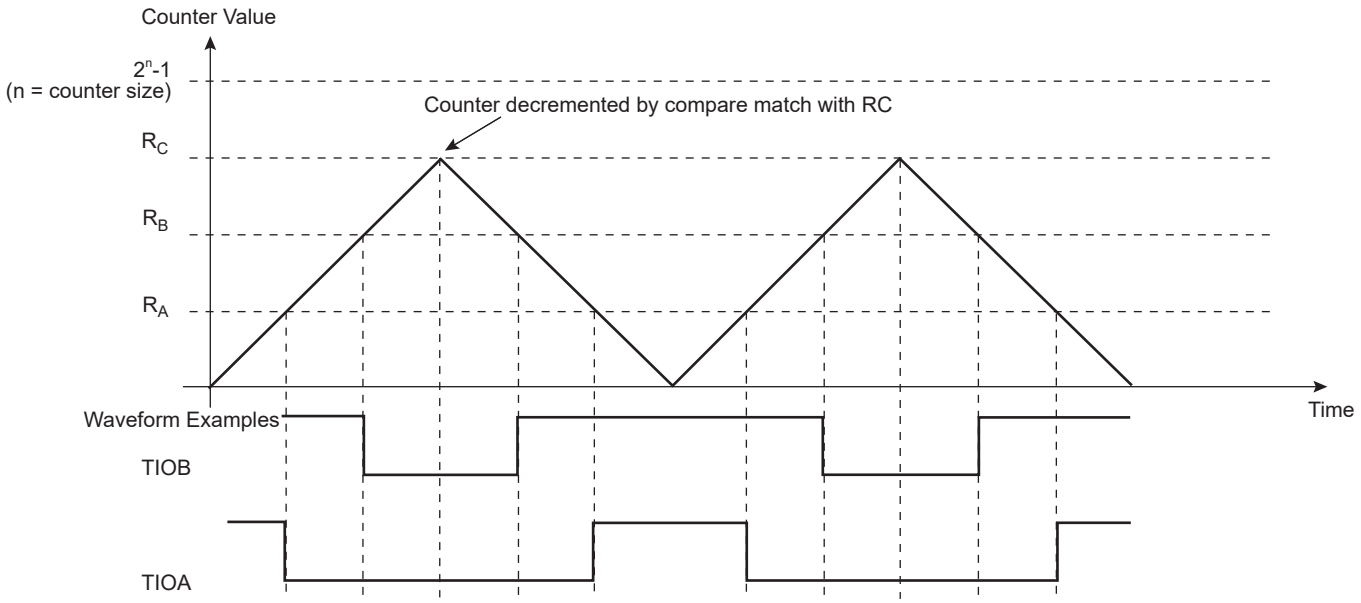
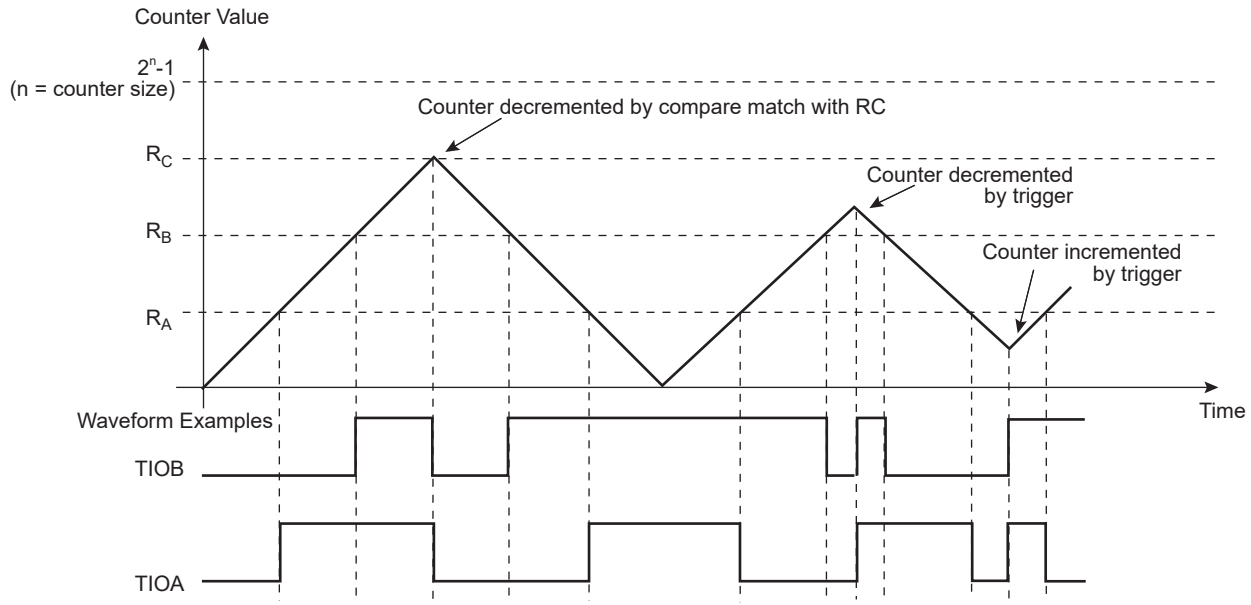


Figure 51-15. WAVSEL = 11 with Trigger



51.6.13 External Event/Trigger Conditions

An external event can be programmed to be detected on one of the clock sources (XC0, XC1, XC2) or TIOBx. The external event selected can then be used as a trigger.

The EEVT parameter in TC_CMR selects the external trigger. The EEVTEDG parameter defines the trigger edge for each of the possible external triggers (rising, falling or both). If EEVTEDG is cleared (none), no external event is defined.

If TIOBx is defined as an external event signal (EEVT = 0), TIOBx is no longer used as an output and the compare register B is not used to generate waveforms and subsequently no IRQs. In this case the TC channel can only generate a waveform on TIOAx.

When an external event is defined, it can be used as a trigger by setting bit ENETRG in the TC_CMR.

As in Capture mode, the SYNC signal and the software trigger are also available as triggers. RC Compare can also be used as a trigger depending on the parameter WAVSEL.

51.6.14 Synchronization with PWM

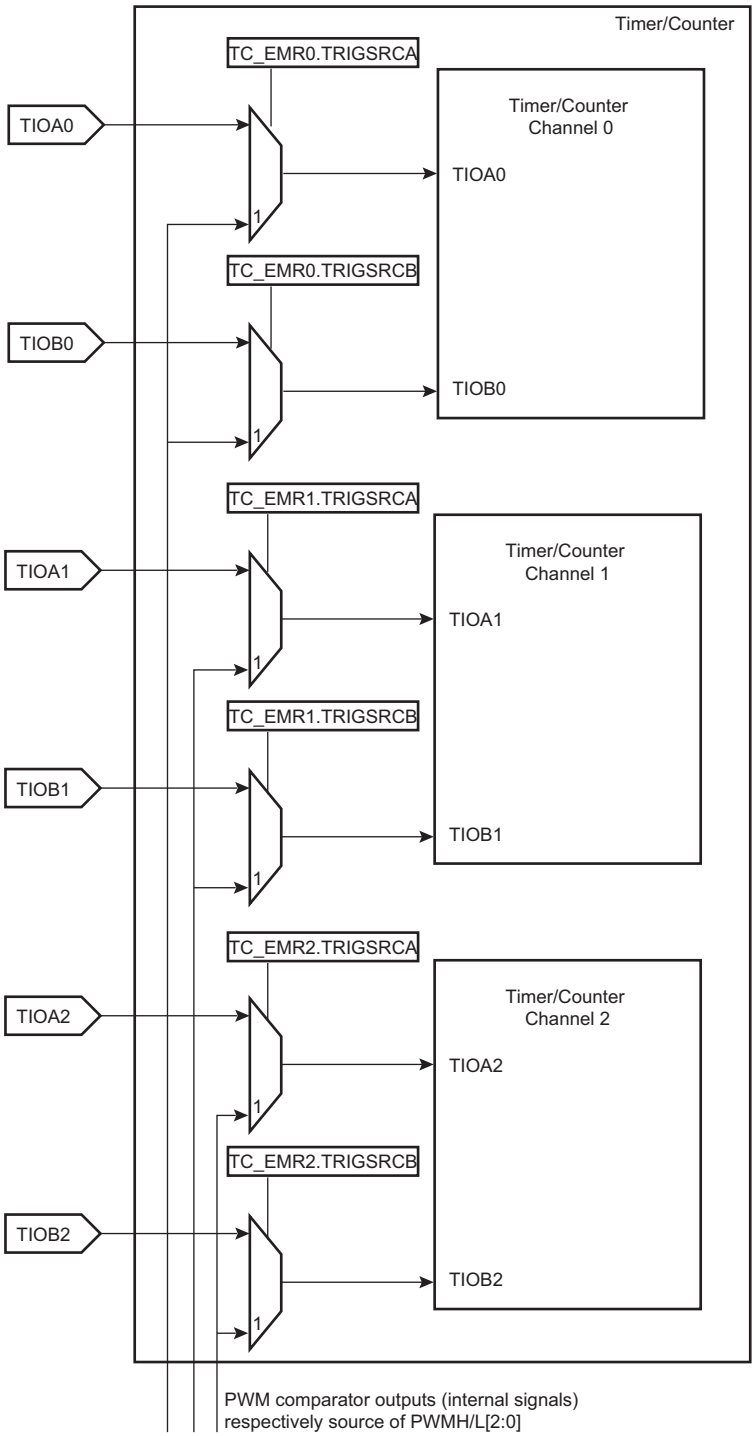
The inputs TIOAx/TIOBx can be bypassed, and thus channel trigger/capture events can be directly driven by the independent PWM module.

PWM comparator outputs (internal signals without dead-time insertion - OCx), respectively source of the PWMH/L[2:0] outputs, are routed to the internal TC inputs. These specific TC inputs are multiplexed with TIOA/B input signal to drive the internal trigger/capture events.

The selection can be programmed in the Extended Mode Register (TC_EMR) fields TRIGSRCA and TRIGSRCB (see Section 51.7.14 “TC Extended Mode Register”).

Each channel of the TC module can be synchronized by a different PWM channel as described in Figure 51-16.

Figure 51-16. Synchronization with PWM



51.6.15 Output Controller

The output controller defines the output level changes on TIOAx and TIOBx following an event. TIOBx Control is used only if TIOBx is defined as output (not as an external event).

The following events control TIOAx and TIOBx:

- Software Trigger
- External Event
- RC Compare

RA Compare controls TIOAx, and RB Compare controls TIOBx. Each of these events can be programmed to set, clear or toggle the output as defined in the corresponding parameter in TC_CMR.

51.6.16 Quadrature Decoder

51.6.16.1 Description

The quadrature decoder (QDEC) is driven by TIOA0, TIOB0, TIOB1 input pins and drives the timer/counter of channel 0 and 1. Channel 2 can be used as a time base in case of speed measurement requirements (refer to Figure 51-17).

When writing a 0 to bit QDEN of the TC_BMR, the QDEC is bypassed and the IO pins are directly routed to the timer counter function.

TIOA0 and TIOB0 are to be driven by the two dedicated quadrature signals from a rotary sensor mounted on the shaft of the off-chip motor.

A third signal from the rotary sensor can be processed through pin TIOB1 and is typically dedicated to be driven by an index signal if it is provided by the sensor. This signal is not required to decode the quadrature signals PHA, PHB.

Field TCCLKS of TC_CMRx must be configured to select XC0 input (i.e., 0x101). Field TC0XC0S has no effect as soon as the QDEC is enabled.

Either speed or position/revolution can be measured. Position channel 0 accumulates the edges of PHA, PHB input signals giving a high accuracy on motor position whereas channel 1 accumulates the index pulses of the sensor, therefore the number of rotations. Concatenation of both values provides a high level of precision on motion system position.

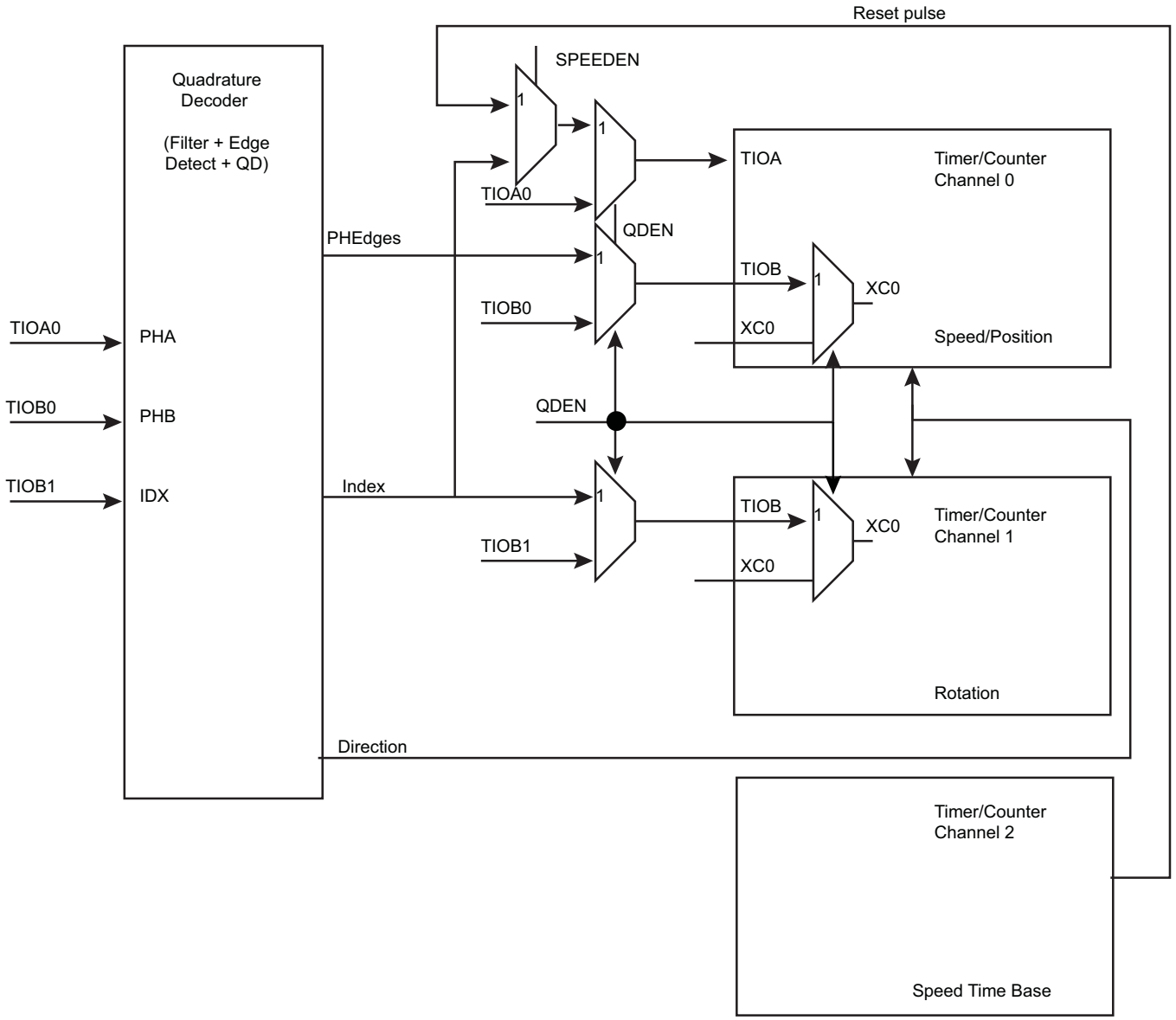
In Speed mode, position cannot be measured but revolution can be measured.

Inputs from the rotary sensor can be filtered prior to down-stream processing. Accommodation of input polarity, phase definition and other factors are configurable.

Interruptions can be generated on different events.

A compare function (using TC_RC) is available on channel 0 (speed/position) or channel 1 (rotation) and can generate an interrupt by means of the CPCS flag in the TC_SRx.

Figure 51-17. Predefined Connection of the Quadrature Decoder with Timer Counters



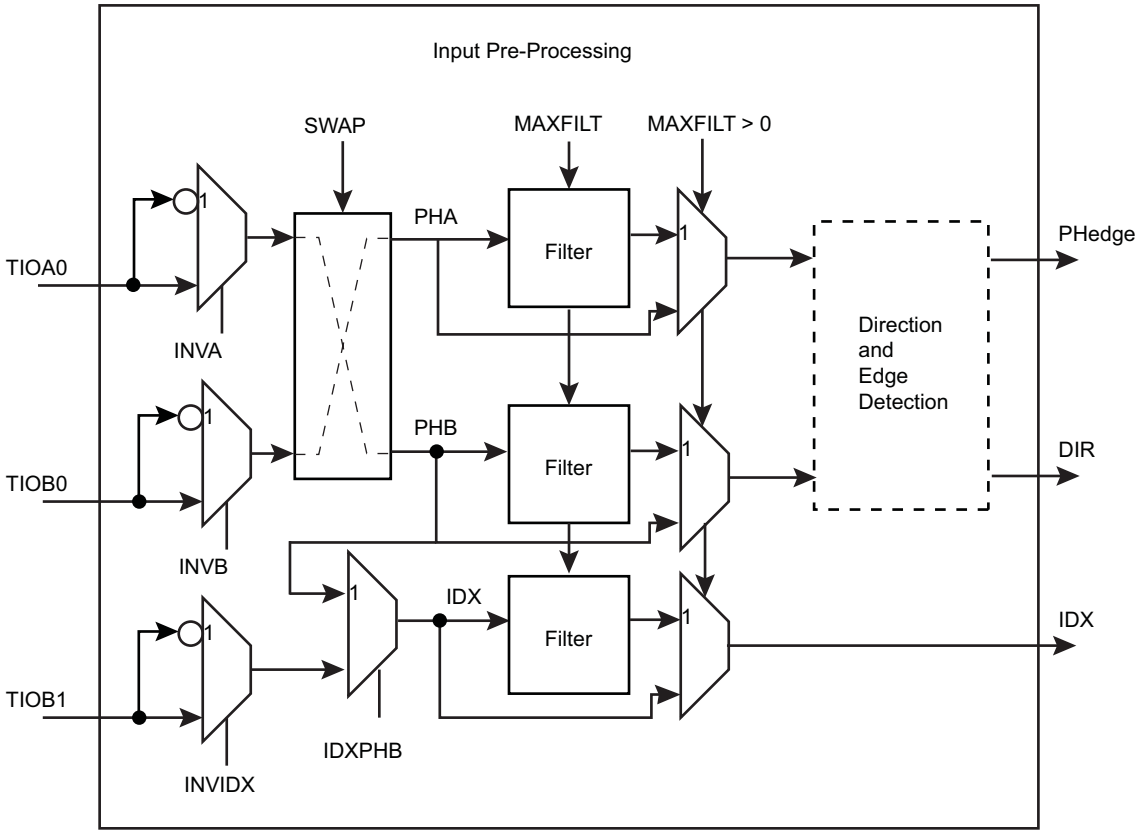
51.6.16.2 Input Preprocessing

Input preprocessing consists of capabilities to take into account rotary sensor factors such as polarities and phase definition followed by configurable digital filtering.

Each input can be negated and swapping PHA, PHB is also configurable.

The MAXFILT field in the TC_BMR is used to configure a minimum duration for which the pulse is stated as valid. When the filter is active, pulses with a duration lower than MAXFILT +1 $\times$ $t_{peripheral\ clock}$ ns are not passed to down-stream logic.

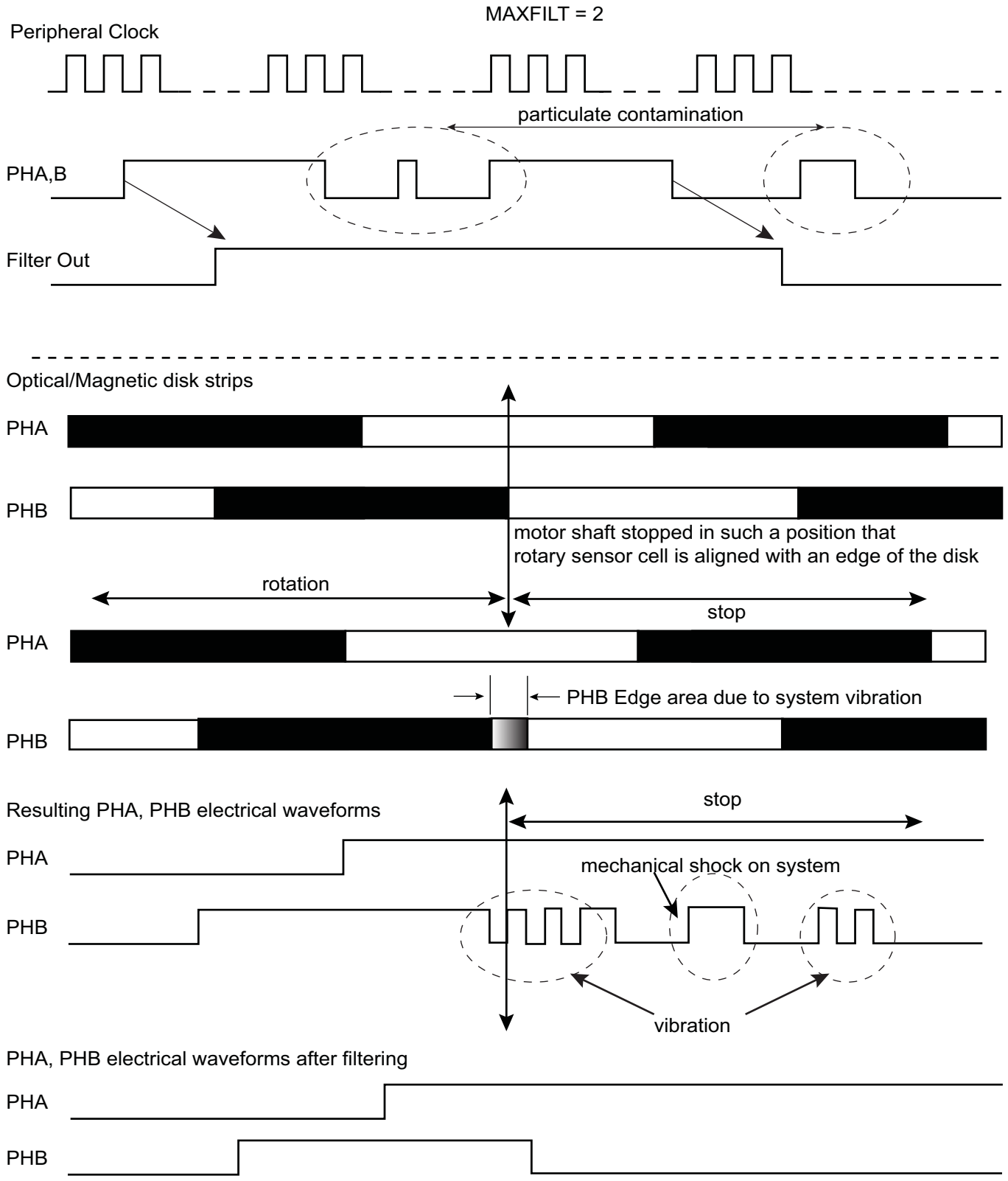
Figure 51-18. Input Stage



Input filtering can efficiently remove spurious pulses that might be generated by the presence of particulate contamination on the optical or magnetic disk of the rotary sensor.

Spurious pulses can also occur in environments with high levels of electro-magnetic interference. Or, simply if vibration occurs even when rotation is fully stopped and the shaft of the motor is in such a position that the beginning of one of the reflective or magnetic bars on the rotary sensor disk is aligned with the light or magnetic (Hall) receiver cell of the rotary sensor. Any vibration can make the PHA, PHB signals toggle for a short duration.

Figure 51-19. Filtering Examples



51.6.16.3 Direction Status and Change Detection

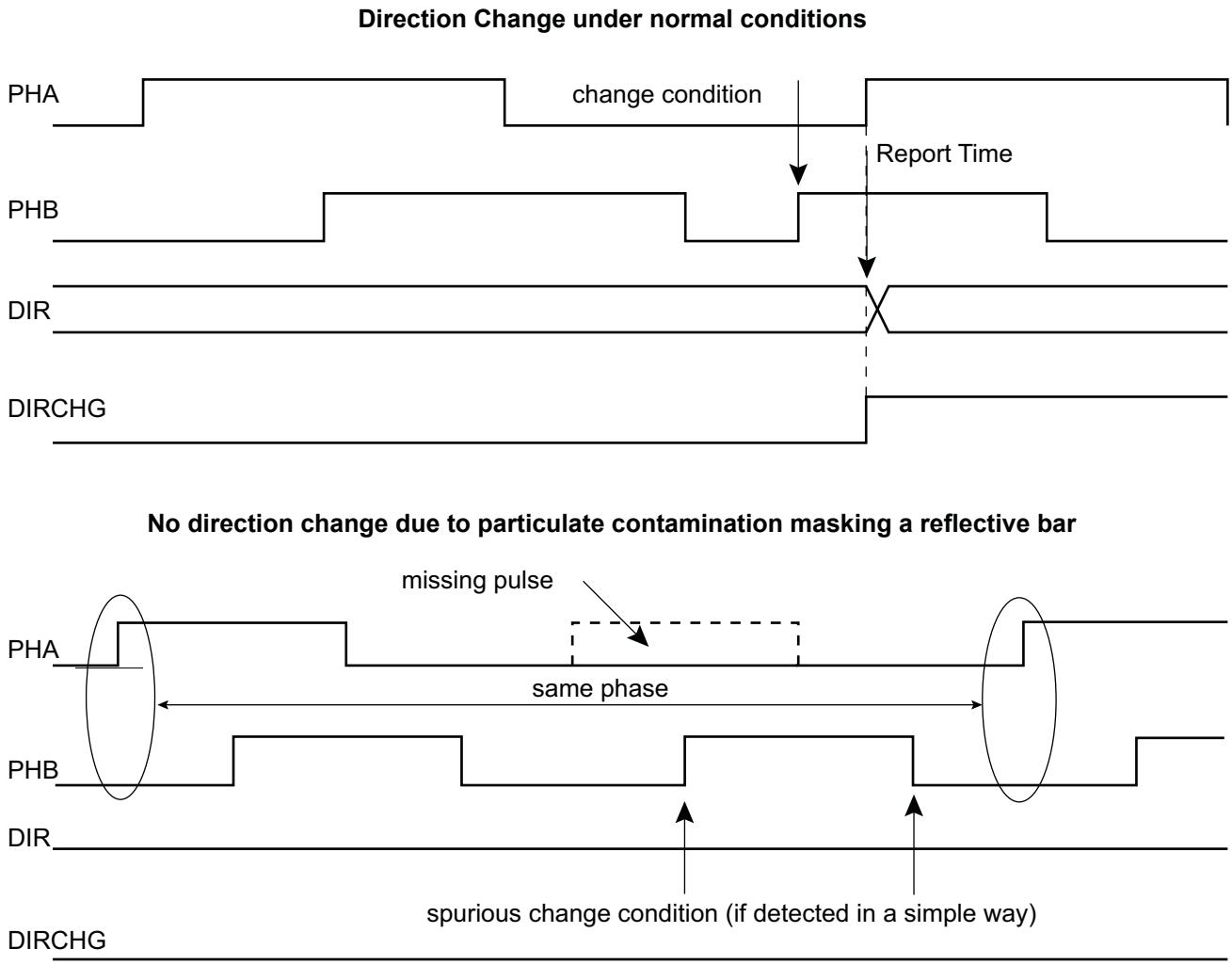
After filtering, the quadrature signals are analyzed to extract the rotation direction and edges of the two quadrature signals detected in order to be counted by timer/counter logic downstream.

The direction status can be directly read at anytime in the TC_QISR. The polarity of the direction flag status depends on the configuration written in TC_BMR. INVA, INVB, INVIDX, SWAP modify the polarity of DIR flag.

Any change in rotation direction is reported in the TC_QISR and can generate an interrupt.

The direction change condition is reported as soon as two consecutive edges on a phase signal have sampled the same value on the other phase signal and there is an edge on the other signal. The two consecutive edges of one phase signal sampling the same value on other phase signal is not sufficient to declare a direction change, for the reason that particulate contamination may mask one or more reflective bars on the optical or magnetic disk of the sensor. Refer to Figure 51-20 for waveforms.

Figure 51-20. Rotation Change Detection

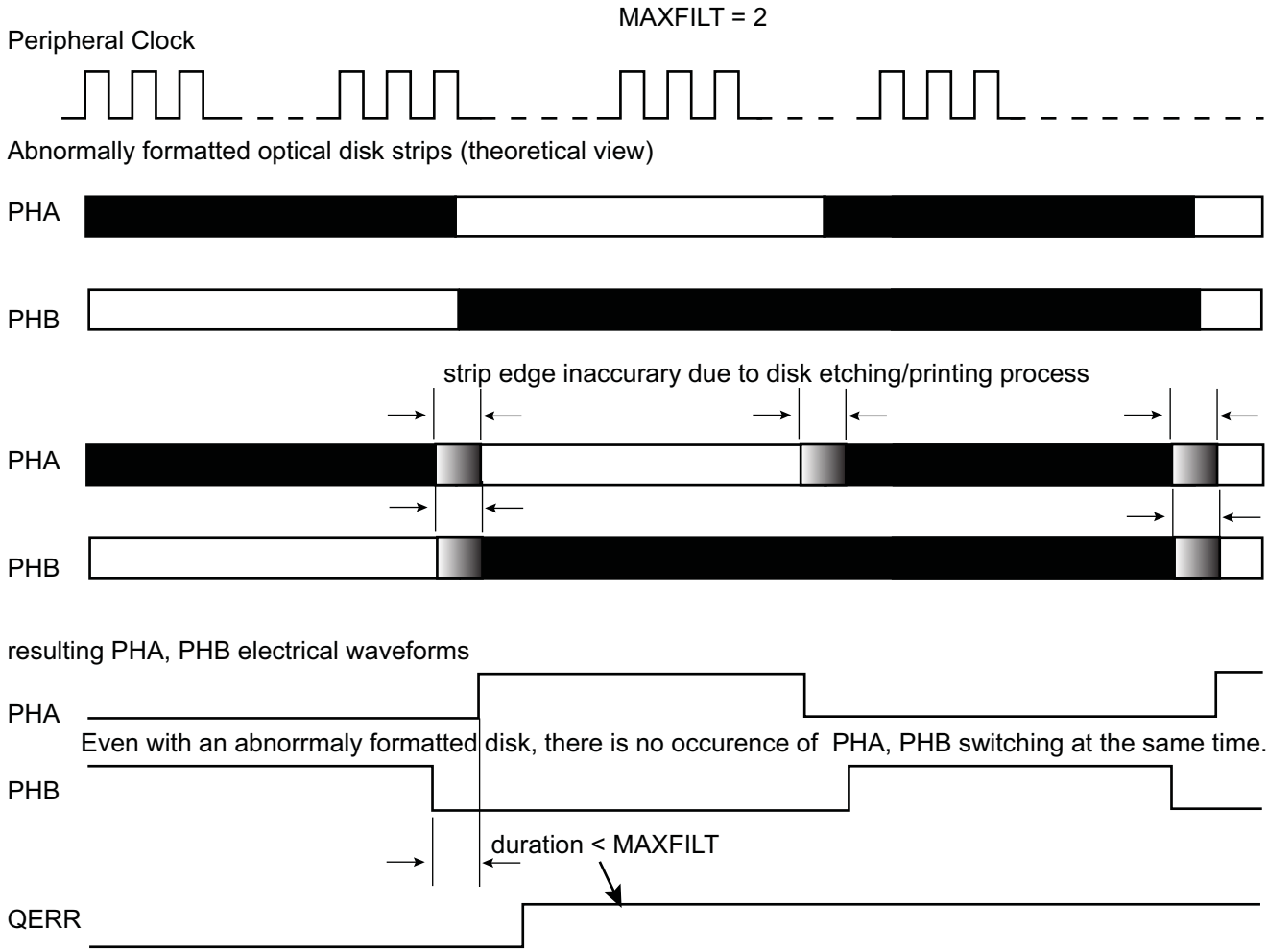


The direction change detection is disabled when QDTRANS is set in the TC_BMR. In this case, the DIR flag report must not be used.

A quadrature error is also reported by the QDEC via the QERR flag in the TC_QISR. This error is reported if the time difference between two edges on PHA, PHB is lower than a predefined value. This predefined value is

configurable and corresponds to (MAXFILT + 1) × $t_{peripheral\ clock}$ ns. After being filtered there is no reason to have two edges closer than (MAXFILT + 1) × $t_{peripheral\ clock}$ ns under normal mode of operation.

Figure 51-21. Quadrature Error Detection



MAXFILT must be tuned according to several factors such as the peripheral clock frequency, type of rotary sensor and rotation speed to be achieved.

51.6.16.4 Position and Rotation Measurement

When the POSEN bit is set in the TC_BMR, the motor axis position is processed on channel 0 (by means of the PHA, PHB edge detections) and the number of motor revolutions are recorded on channel 1 if the IDX signal is provided on the TIOB1 input. If there is no IDX signal available, it is possible to clear the internal counter for each revolution if the number of counts per revolution is configured in TC_RC0.RC and the TC_CMR.CPCTRG bit is written to 1. The position measurement can be read in the TC_CV0 register and the rotation measurement can be read in the TC_CV1 register.

Channel 0 and 1 must be configured in Capture mode (TC_CMR0.WAVE = 0). ‘Rising edge’ must be selected as the External Trigger Edge (TC_CMR.ETRGEDG = 0x01) and ‘TIOAx’ must be selected as the External Trigger (TC_CMR.ABETRG = 0x1).

In parallel, the number of edges are accumulated on timer/counter channel 0 and can be read on the TC_CV0 register.

Therefore, the accurate position can be read on both TC_CV registers and concatenated to form a 32-bit word.

The timer/counter channel 0 is cleared for each increment of IDX count value.

Depending on the quadrature signals, the direction is decoded and allows to count up or down in timer/counter channels 0 and 1. The direction status is reported on TC_QISR.

51.6.16.5 Speed Measurement

When SPEEDEN is set in the TC_BMR, the speed measure is enabled on channel 0.

A time base must be defined on channel 2 by writing the TC_RC2 period register. Channel 2 must be configured in Waveform mode (WAVE bit set) in TC_CMR2. The WAVSEL field must be defined with 0x10 to clear the counter by comparison and matching with TC_RC value. Field ACPC must be defined at 0x11 to toggle TIOAx output.

This time base is automatically fed back to TIOAx of channel 0 when QDEN and SPEEDEN are set.

Channel 0 must be configured in Capture mode (WAVE = 0 in TC_CMR0). The ABETRG bit of TC_CMR0 must be configured at 1 to select TIOAx as a trigger for this channel.

EDGTRG must be set to 0x01, to clear the counter on a rising edge of the TIOAx signal and field LDRA must be set accordingly to 0x01, to load TC_RA0 at the same time as the counter is cleared (LDRB must be set to 0x01). As a consequence, at the end of each time base period the differentiation required for the speed calculation is performed.

The process must be started by configuring bits CLKEN and SWTRG in the TC_CCR.

The speed can be read on field RA in TC_RA0.

Channel 1 can still be used to count the number of revolutions of the motor.

51.6.16.6 Detecting a Missing Index Pulse

To detect a missing index pulse due contamination, dust, etc., the TC_SR0.CPCS flag can be used. It is also possible to assert the interrupt line if the TC_SR0.CPCS flag is enabled as a source of the interrupt by writing a ‘1’ to TC_IER0.CPCS.

The TC_RC0.RC field must be written with the nominal number of counts per revolution provided by the rotary encoder, plus a margin to eliminate potential noise (e.g., if nominal count per revolution is 1024, then TC_RC0.RC = 1028).

If the index pulse is missing, the timer value is not cleared and the nominal value is exceeded, then the comparator on the RC triggers an event, TC_SR0.CPCS = 1, and the interrupt line is asserted if TC_IER0.CPCS = 1.

51.6.16.7 Missing Pulse Detection and Auto-correction

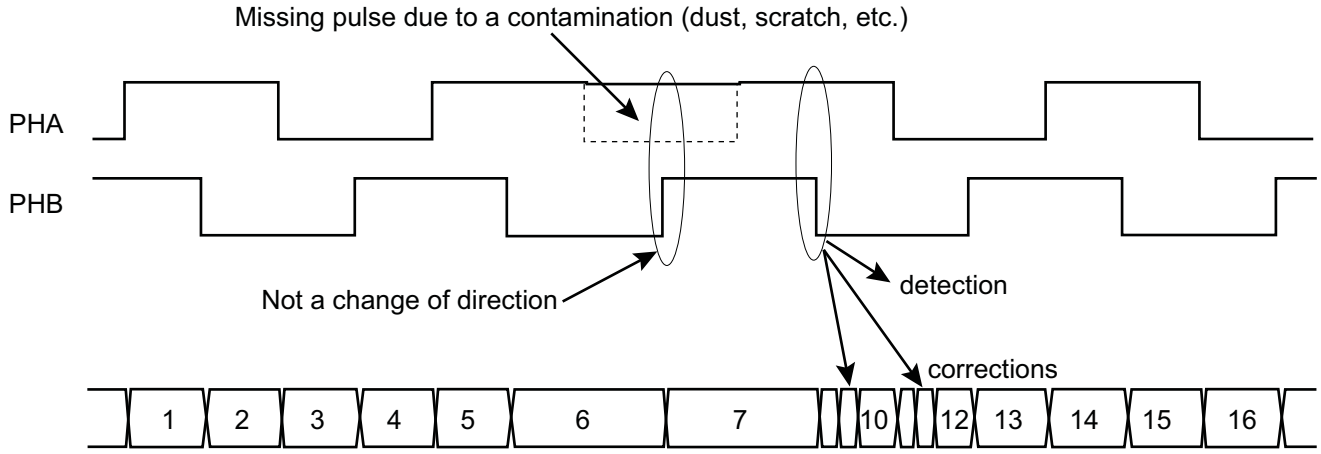
The QDEC is equipped with a circuitry which detects and corrects some errors that may result from contamination on optical disks or other materials producing the quadrature phase signals.

The detection and autocorrection only works if the Count mode is configured for both phases (EDGPHA = 1 in TC_BMR) and is enabled (AUTOC = 1 in TC_BMR).

If a pulse is missing on a phase signal, it is automatically detected and the pulse count reported in the CV field of the TC_CV0/1 is automatically corrected.

There is no detection if both phase signals are affected at the same location on the device providing the quadrature signals because the detection requires a valid phase signal to detect the contamination on the other phase signal.

Figure 51-22. Detection and Auto-correction of Missing Pulses



If a quadrature device is undamaged, the number of pulses counted for a predefined period of time must be the same with or without detection and auto-correction feature.

Therefore, if the measurement results differ, a contamination exists on the device producing the quadrature signals.

This does not substitute the measurements of the number of pulses between two index pulses (if available) but provides a complementary method to detect damaged quadrature devices.

When the device providing quadrature signals is severely damaged, potentially leading to a number of consecutive missing pulses greater than 1, the downstream processing may be affected. It is possible to define the maximum admissible number of consecutive missing pulses before issuing a Missing Pulse Error flag (MPE in TC_QISR). The threshold triggering a MPE flag report can be configured in field MAXCMP of the TC_BMR. If the field MAXCMP is cleared, MPE never rises. The flag MAXCMP can trigger an interrupt while the QDEC is operating, thus providing a real time report of a potential problem on the quadrature device.

51.6.17 2-bit Gray Up/Down Counter for Stepper Motor

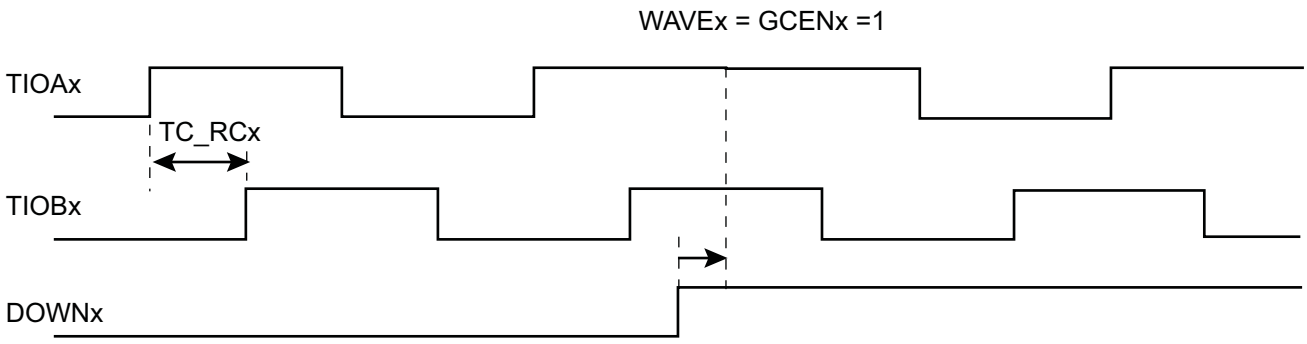
Each channel can be independently configured to generate a 2-bit Gray count waveform on corresponding TIOAx, TIOBx outputs by means of the GCEN bit in TC_SMMRx.

Up or Down count can be defined by writing bit DOWN in TC_SMMRx.

It is mandatory to configure the channel in Waveform mode in the TC_CMR.

The period of the counters can be programmed in TC_RCx.

Figure 51-23. 2-bit Gray Up/Down Counter



51.6.18 Fault Mode

At any time, the TC_RCx registers can be used to perform a comparison on the respective current channel counter value (TC_CVx) with the value of TC_RCx register.

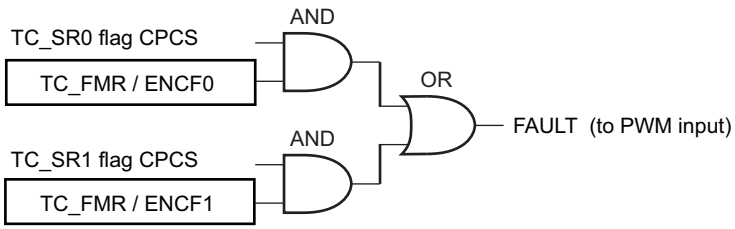
The CPCSx flags can be set accordingly and an interrupt can be generated.

This interrupt is processed but requires an unpredictable amount of time to be achieve the required action.

It is possible to trigger the FAULT output of the TIMER1 with CPCS from TC_SR0 and/or CPCS from TC_SR1. Each source can be independently enabled/disabled in the TC_FMR.

This can be useful to detect an overflow on speed and/or position when QDEC is processed and to act immediately by using the FAULT output.

Figure 51-24. Fault Output Generation



51.6.19 Register Write Protection

To prevent any single software error from corrupting TC behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the TC Write Protection Mode Register (TC_WPMR).

The Timer Counter clock of the first channel must be enabled to access TC_WPMR.

The following registers can be write-protected:

- TC Block Mode Register
- TC Channel Mode Register: Capture Mode
- TC Channel Mode Register: Waveform Mode
- TC Fault Mode Register
- TC Stepper Motor Mode Register
- TC Register A
- TC Register B
- TC Register C
- TC Extended Mode Register

51.7 Timer Counter (TC) User Interface

Table 51-6. Register Mapping

Offset[1]	Register	Name	Access	Reset
0x00 + channel * 0x40 + 0x00	Channel Control Register	TC_CCR	Write-only	–
0x00 + channel * 0x40 + 0x04	Channel Mode Register	TC_CMR	Read/Write	0
0x00 + channel * 0x40 + 0x08	Stepper Motor Mode Register	TC_SMMR	Read/Write	0
0x00 + channel * 0x40 + 0x0C	Register AB	TC_RAB	Read-only	0
0x00 + channel * 0x40 + 0x10	Counter Value	TC_CV	Read-only	0
0x00 + channel * 0x40 + 0x14	Register A	TC_RA	Read/Write[2]	0
0x00 + channel * 0x40 + 0x18	Register B	TC_RB	Read/Write[2]	0
0x00 + channel * 0x40 + 0x1C	Register C	TC_RC	Read/Write	0
0x00 + channel * 0x40 + 0x20	Status Register	TC_SR	Read-only	0
0x00 + channel * 0x40 + 0x24	Interrupt Enable Register	TC_IER	Write-only	–
0x00 + channel * 0x40 + 0x28	Interrupt Disable Register	TC_IDR	Write-only	–
0x00 + channel * 0x40 + 0x2C	Interrupt Mask Register	TC_IMR	Read-only	0
0x00 + channel * 0x40 + 0x30	Extended Mode Register	TC_EMR	Read/Write	0
0xC0	Block Control Register	TC_BCR	Write-only	–
0xC4	Block Mode Register	TC_BMR	Read/Write	0
0xC8	QDEC Interrupt Enable Register	TC_QIER	Write-only	–
0xCC	QDEC Interrupt Disable Register	TC_QIDR	Write-only	–
0xD0	QDEC Interrupt Mask Register	TC_QIMR	Read-only	0
0xD4	QDEC Interrupt Status Register	TC_QISR	Read-only	0
0xD8	Fault Mode Register	TC_FMR	Read/Write	0
0xE4	Write Protection Mode Register	TC_WPMR	Read/Write	0
0xE8–0xFC	Reserved	–	–	–

Notes: 1. Channel index ranges from 0 to 2.
2. Read-only if TC_CMRx.WAVE = 0

51.7.1 TC Channel Control Register

Name: TC_CCRx [x=0..2]

Address: 0xF800C000 (0)[0], 0xF800C040 (0)[1], 0xF800C080 (0)[2], 0xF8010000 (1)[0], 0xF8010040 (1)[1], 0xF8010080 (1)[2]

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	SWTRG	CLKDIS	CLKEN

- **CLKEN: Counter Clock Enable Command**

0: No effect.

1: Enables the clock if CLKDIS is not 1.

- **CLKDIS: Counter Clock Disable Command**

0: No effect.

1: Disables the clock.

- **SWTRG: Software Trigger Command**

0: No effect.

1: A software trigger is performed: the counter is reset and the clock is started.

51.7.2 TC Channel Mode Register: Capture Mode

Name: TC_CMRx [x=0..2] (CAPTURE_MODE)

Address: 0xF800C004 (0)[0], 0xF800C044 (0)[1], 0xF800C084 (0)[2], 0xF8010004 (1)[0], 0xF8010044 (1)[1], 0xF8010084 (1)[2]

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	SBSMPLR			LDRB		LDRA	

15	14	13	12	11	10	9	8
WAVE	CPCTRG	–	–	–	ABETRG	ETRGEDG	

7	6	5	4	3	2	1	0
LDBDIS	LDBSTOP	BURST		CLKI	TCCLKS		

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

- **TCCLKS: Clock Selection**

Value	Name	Description
0	TIMER_CLOCK1	Clock selected: internal GCLK [35], GCLK [36] clock signal (from PMC)
1	TIMER_CLOCK2	Clock selected: internal System bus clock divided by 8 clock signal (from PMC)
2	TIMER_CLOCK3	Clock selected: internal System bus clock divided by 32 clock signal (from PMC)
3	TIMER_CLOCK4	Clock selected: internal System bus clock divided by 128 clock signal (from PMC)
4	TIMER_CLOCK5	Clock selected: internal slow_clock clock signal (from PMC)
5	XC0	Clock selected: XC0
6	XC1	Clock selected: XC1
7	XC2	Clock selected: XC2

To operate at maximum peripheral clock frequency, refer to Section 51.7.14 “TC Extended Mode Register”.

- **CLKI: Clock Invert**

0: Counter is incremented on rising edge of the clock.

1: Counter is incremented on falling edge of the clock.

- **BURST: Burst Signal Selection**

Value	Name	Description
0	NONE	The clock is not gated by an external signal.
1	XC0	XC0 is ANDed with the selected clock.
2	XC1	XC1 is ANDed with the selected clock.
3	XC2	XC2 is ANDed with the selected clock.

- **LDBSTOP: Counter Clock Stopped with RB Loading**

0: Counter clock is not stopped when RB loading occurs.

1: Counter clock is stopped when RB loading occurs.

- **LDBDIS: Counter Clock Disable with RB Loading**

0: Counter clock is not disabled when RB loading occurs.

1: Counter clock is disabled when RB loading occurs.

- **ETRGEDG: External Trigger Edge Selection**

Value	Name	Description
0	NONE	The clock is not gated by an external signal.
1	RISING	Rising edge
2	FALLING	Falling edge
3	EDGE	Each edge

- **ABETRG: TIOAx or TIOBx External Trigger Selection**

0: TIOBx is used as an external trigger.

1: TIOAx is used as an external trigger.

- **CPCTRG: RC Compare Trigger Enable**

0: RC Compare has no effect on the counter and its clock.

1: RC Compare resets the counter and starts the counter clock.

- **WAVE: Waveform Mode**

0: Capture mode is enabled.

1: Capture mode is disabled (Waveform mode is enabled).

- **LDRA: RA Loading Edge Selection**

Value	Name	Description
0	NONE	None
1	RISING	Rising edge of TIOAx
2	FALLING	Falling edge of TIOAx
3	EDGE	Each edge of TIOAx

- **LDRB: RB Loading Edge Selection**

Value	Name	Description
0	NONE	None
1	RISING	Rising edge of TIOAx
2	FALLING	Falling edge of TIOAx
3	EDGE	Each edge of TIOAx

- **SBSMPLR: Loading Edge Subsampling Ratio**

Value	Name	Description
0	ONE	Load a Capture Register each selected edge
1	HALF	Load a Capture Register every 2 selected edges
2	FOURTH	Load a Capture Register every 4 selected edges
3	EIGHTH	Load a Capture Register every 8 selected edges
4	SIXTEENTH	Load a Capture Register every 16 selected edges

51.7.3 TC Channel Mode Register: Waveform Mode

Name: TC_CMRx [x=0..2] (WAVEFORM_MODE)

Address: 0xF800C004 (0)[0], 0xF800C044 (0)[1], 0xF800C084 (0)[2], 0xF8010004 (1)[0], 0xF8010044 (1)[1], 0xF8010084 (1)[2]

Access: Read/Write

31	30	29	28	27	26	25	24
BSWTRG		BEEVT		BCPC		BCPB	
23	22	21	20	19	18	17	16
ASWTRG		AEEVT		ACPC		ACPA	
15	14	13	12	11	10	9	8
WAVE	WAVSEL		ENETRG	EEVT		EEVTEDG	
7	6	5	4	3	2	1	0
CPCDIS	CPCSTOP	BURST		CLKI	TCCLKS		

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

- **TCCLKS: Clock Selection**

Value	Name	Description
0	TIMER_CLOCK1	Clock selected: internal GCLK [35], GCLK [36] clock signal (from PMC)
1	TIMER_CLOCK2	Clock selected: internal System bus clock divided by 8 clock signal (from PMC)
2	TIMER_CLOCK3	Clock selected: internal System bus clock divided by 32 clock signal (from PMC)
3	TIMER_CLOCK4	Clock selected: internal System bus clock divided by 128 clock signal (from PMC)
4	TIMER_CLOCK5	Clock selected: internal slow_clock clock signal (from PMC)
5	XC0	Clock selected: XC0
6	XC1	Clock selected: XC1
7	XC2	Clock selected: XC2

To operate at maximum peripheral clock frequency, refer to Section 51.7.14 “TC Extended Mode Register”.

- **CLKI: Clock Invert**

0: Counter is incremented on rising edge of the clock.

1: Counter is incremented on falling edge of the clock.

- **BURST: Burst Signal Selection**

Value	Name	Description
0	NONE	The clock is not gated by an external signal.
1	XC0	XC0 is ANDed with the selected clock.
2	XC1	XC1 is ANDed with the selected clock.
3	XC2	XC2 is ANDed with the selected clock.

- **CPCSTOP: Counter Clock Stopped with RC Compare**

0: Counter clock is not stopped when counter reaches RC.

1: Counter clock is stopped when counter reaches RC.

- **CPCDIS: Counter Clock Disable with RC Compare**

0: Counter clock is not disabled when counter reaches RC.

1: Counter clock is disabled when counter reaches RC.

- **EEVTEDG: External Event Edge Selection**

Value	Name	Description
0	NONE	None
1	RISING	Rising edge
2	FALLING	Falling edge
3	EDGE	Each edge

- **EEVT: External Event Selection**

Signal selected as external event.

Value	Name	Description	TIOB Direction
0	TIOB	TIOB(1)	Input
1	XC0	XC0	Output
2	XC1	XC1	Output
3	XC2	XC2	Output

Note: 1. If TIOB is chosen as the external event signal, it is configured as an input and no longer generates waveforms and subsequently no IRQs.

- **ENETRG: External Event Trigger Enable**

0: The external event has no effect on the counter and its clock.

1: The external event resets the counter and starts the counter clock.

Note: Whatever the value programmed in ENETRG, the selected external event only controls the TIOAx output and TIOBx if not used as input (trigger event input or other input used).

- **WAVSEL: Waveform Selection**

Value	Name	Description
0	UP	UP mode without automatic trigger on RC Compare
1	UPDOWN	UPDOWN mode without automatic trigger on RC Compare
2	UP_RC	UP mode with automatic trigger on RC Compare
3	UPDOWN_RC	UPDOWN mode with automatic trigger on RC Compare

- **WAVE: Waveform Mode**

0: Waveform mode is disabled (Capture mode is enabled).

1: Waveform mode is enabled.

- **ACPA: RA Compare Effect on TIOAx**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

- **ACPC: RC Compare Effect on TIOAx**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

- **AEEVT: External Event Effect on TIOAx**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

- **ASWTRG: Software Trigger Effect on TIOAx**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

- **BCPB: RB Compare Effect on TIOBx**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

- **BCPC: RC Compare Effect on TIOBx**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

- **BEEVT: External Event Effect on TIOBx**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

- **BSWTRG: Software Trigger Effect on TIOBx**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

51.7.4 TC Stepper Motor Mode Register

Name: TC_SMMRx [x=0..2]

Address: 0xF800C008 (0)[0], 0xF800C048 (0)[1], 0xF800C088 (0)[2], 0xF8010008 (1)[0], 0xF8010048 (1)[1], 0xF8010088 (1)[2]

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	DOWN	GCEN

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

- **GCEN: Gray Count Enable**

0: TIOAx [x=0..2] and TIOBx [x=0..2] are driven by internal counter of channel x.

1: TIOAx [x=0..2] and TIOBx [x=0..2] are driven by a 2-bit Gray counter.

- **DOWN: Down Count**

0: Up counter.

1: Down counter.

51.7.5 TC Register AB

Name: TC_RABx [x=0..2]

Address: 0xF800C00C (0)[0], 0xF800C04C (0)[1], 0xF800C08C (0)[2], 0xF801000C (1)[0], 0xF801004C (1)[1], 0xF801008C (1)[2]

Access: Read-only

31	30	29	28	27	26	25	24
RAB							

23	22	21	20	19	18	17	16
RAB							

15	14	13	12	11	10	9	8
RAB							

7	6	5	4	3	2	1	0
RAB							

- **RAB: Register A or Register B**

RAB contains the next unread capture Register A or Register B value in real time. It is usually read by the DMA after a request due to a valid load edge on TIOAx.

When DMA is used, the RAB register address must be configured as source address of the transfer.

51.7.6 TC Counter Value Register

Name: TC_CVx [x=0..2]

Address: 0xF800C010 (0)[0], 0xF800C050 (0)[1], 0xF800C090 (0)[2], 0xF8010010 (1)[0], 0xF8010050 (1)[1], 0xF8010090 (1)[2]

Access: Read-only

31	30	29	28	27	26	25	24
CV							

23	22	21	20	19	18	17	16
CV							

15	14	13	12	11	10	9	8
CV							

7	6	5	4	3	2	1	0
CV							

- **CV: Counter Value**

CV contains the counter value in real time.

51.7.7 TC Register A

Name: TC_RAx [x=0..2]

Address: 0xF800C014 (0)[0], 0xF800C054 (0)[1], 0xF800C094 (0)[2], 0xF8010014 (1)[0], 0xF8010054 (1)[1], 0xF8010094 (1)[2]

Access: Read-only if TC_CMRx.WAVE = 0, Read/Write if TC_CMRx.WAVE = 1

31	30	29	28	27	26	25	24
RA							
23	22	21	20	19	18	17	16
RA							
15	14	13	12	11	10	9	8
RA							
7	6	5	4	3	2	1	0
RA							

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

- **RA: Register A**

RA contains the Register A value in real time.

51.7.8 TC Register B

Name: TC_RBx [x=0..2]

Address: 0xF800C018 (0)[0], 0xF800C058 (0)[1], 0xF800C098 (0)[2], 0xF8010018 (1)[0], 0xF8010058 (1)[1], 0xF8010098 (1)[2]

Access: Read-only if TC_CMRx.WAVE = 0, Read/Write if TC_CMRx.WAVE = 1

31	30	29	28	27	26	25	24
RB							
23	22	21	20	19	18	17	16
RB							
15	14	13	12	11	10	9	8
RB							
7	6	5	4	3	2	1	0
RB							

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

- **RB: Register B**

RB contains the Register B value in real time.

51.7.9 TC Register C

Name: TC_RCx [x=0..2]

Address: 0xF800C01C (0)[0], 0xF800C05C (0)[1], 0xF800C09C (0)[2], 0xF801001C (1)[0], 0xF801005C (1)[1], 0xF801009C (1)[2]

Access: Read/Write

31	30	29	28	27	26	25	24
RC							

23	22	21	20	19	18	17	16
RC							

15	14	13	12	11	10	9	8
RC							

7	6	5	4	3	2	1	0
RC							

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

- **RC: Register C**

RC contains the Register C value in real time.

51.7.10 TC Status Register

Name: TC_SRx [x=0..2]

Address: 0xF800C020 (0)[0], 0xF800C060 (0)[1], 0xF800C0A0 (0)[2], 0xF8010020 (1)[0], 0xF8010060 (1)[1], 0xF80100A0 (1)[2]

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	MTIOB	MTIOA	CLKSTA
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

- **COVFS: Counter Overflow Status (cleared on read)**

0: No counter overflow has occurred since the last read of the Status Register.

1: A counter overflow has occurred since the last read of the Status Register.

- **LOVRS: Load Overrun Status (cleared on read)**

0: Load overrun has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 1.

1: RA or RB have been loaded at least twice without any read of the corresponding register since the last read of the Status Register, if TC_CMRx.WAVE = 0.

- **CPAS: RA Compare Status (cleared on read)**

0: RA Compare has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 0.

1: RA Compare has occurred since the last read of the Status Register, if TC_CMRx.WAVE = 1.

- **CPBS: RB Compare Status (cleared on read)**

0: RB Compare has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 0.

1: RB Compare has occurred since the last read of the Status Register, if TC_CMRx.WAVE = 1.

- **CPCS: RC Compare Status (cleared on read)**

0: RC Compare has not occurred since the last read of the Status Register.

1: RC Compare has occurred since the last read of the Status Register.

- **LDRAS: RA Loading Status (cleared on read)**

0: RA Load has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 1.

1: RA Load has occurred since the last read of the Status Register, if TC_CMRx.WAVE = 0.

- **LDRBS: RB Loading Status (cleared on read)**

0: RB Load has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 1.

1: RB Load has occurred since the last read of the Status Register, if TC_CMRx.WAVE = 0.

- **ETRGS: External Trigger Status (cleared on read)**

0: External trigger has not occurred since the last read of the Status Register.

1: External trigger has occurred since the last read of the Status Register.

- **CLKSTA: Clock Enabling Status**

0: Clock is disabled.

1: Clock is enabled.

- **MTIOA: TIOAx Mirror**

0: TIOAx is low. If TC_CMRx.WAVE = 0, this means that TIOAx pin is low. If TC_CMRx.WAVE = 1, this means that TIOAx is driven low.

1: TIOAx is high. If TC_CMRx.WAVE = 0, this means that TIOAx pin is high. If TC_CMRx.WAVE = 1, this means that TIOAx is driven high.

- **MTIOB: TIOBx Mirror**

0: TIOBx is low. If TC_CMRx.WAVE = 0, this means that TIOBx pin is low. If TC_CMRx.WAVE = 1, this means that TIOBx is driven low.

1: TIOBx is high. If TC_CMRx.WAVE = 0, this means that TIOBx pin is high. If TC_CMRx.WAVE = 1, this means that TIOBx is driven high.

51.7.11 TC Interrupt Enable Register

Name: TC_IERx [x=0..2]

Address: 0xF800C024 (0)[0], 0xF800C064 (0)[1], 0xF800C0A4 (0)[2], 0xF8010024 (1)[0], 0xF8010064 (1)[1], 0xF80100A4 (1)[2]

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

- **COVFS: Counter Overflow**

0: No effect.

1: Enables the Counter Overflow Interrupt.

- **LOVRS: Load Overrun**

0: No effect.

1: Enables the Load Overrun Interrupt.

- **CPAS: RA Compare**

0: No effect.

1: Enables the RA Compare Interrupt.

- **CPBS: RB Compare**

0: No effect.

1: Enables the RB Compare Interrupt.

- **CPCS: RC Compare**

0: No effect.

1: Enables the RC Compare Interrupt.

- **LDRAS: RA Loading**

0: No effect.

1: Enables the RA Load Interrupt.

- **LDRBS: RB Loading**

0: No effect.

1: Enables the RB Load Interrupt.

- **ETRGS: External Trigger**

0: No effect.

1: Enables the External Trigger Interrupt.

51.7.12 TC Interrupt Disable Register

Name: TC_IDRx [x=0..2]

Address: 0xF800C028 (0)[0], 0xF800C068 (0)[1], 0xF800C0A8 (0)[2], 0xF8010028 (1)[0], 0xF8010068 (1)[1], 0xF80100A8 (1)[2]

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

- **COVFS: Counter Overflow**

0: No effect.

1: Disables the Counter Overflow Interrupt.

- **LOVRS: Load Overrun**

0: No effect.

1: Disables the Load Overrun Interrupt (if TC_CMRx.WAVE = 0).

- **CPAS: RA Compare**

0: No effect.

1: Disables the RA Compare Interrupt (if TC_CMRx.WAVE = 1).

- **CPBS: RB Compare**

0: No effect.

1: Disables the RB Compare Interrupt (if TC_CMRx.WAVE = 1).

- **CPCS: RC Compare**

0: No effect.

1: Disables the RC Compare Interrupt.

- **LDRAS: RA Loading**

0: No effect.

1: Disables the RA Load Interrupt (if TC_CMRx.WAVE = 0).

- **LDRBS: RB Loading**

0: No effect.

1: Disables the RB Load Interrupt (if TC_CMRx.WAVE = 0).

- **ETRGS: External Trigger**

0: No effect.

1: Disables the External Trigger Interrupt.

51.7.13 TC Interrupt Mask Register

Name: TC_IMRx [x=0..2]

Address: 0xF800C02C (0)[0], 0xF800C06C (0)[1], 0xF800C0AC (0)[2], 0xF801002C (1)[0], 0xF801006C (1)[1], 0xF80100AC (1)[2]

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

- **COVFS: Counter Overflow**

0: The Counter Overflow Interrupt is disabled.

1: The Counter Overflow Interrupt is enabled.

- **LOVRS: Load Overrun**

0: The Load Overrun Interrupt is disabled.

1: The Load Overrun Interrupt is enabled.

- **CPAS: RA Compare**

0: The RA Compare Interrupt is disabled.

1: The RA Compare Interrupt is enabled.

- **CPBS: RB Compare**

0: The RB Compare Interrupt is disabled.

1: The RB Compare Interrupt is enabled.

- **CPCS: RC Compare**

0: The RC Compare Interrupt is disabled.

1: The RC Compare Interrupt is enabled.

- **LDRAS: RA Loading**

0: The Load RA Interrupt is disabled.

1: The Load RA Interrupt is enabled.

- **LDRBS: RB Loading**

0: The Load RB Interrupt is disabled.

1: The Load RB Interrupt is enabled.

- **ETRGS: External Trigger**

0: The External Trigger Interrupt is disabled.

1: The External Trigger Interrupt is enabled.

51.7.14 TC Extended Mode Register

Name: TC_EMRx [x=0..2]

Address: 0xF800C030 (0)[0], 0xF800C070 (0)[1], 0xF800C0B0 (0)[2], 0xF8010030 (1)[0], 0xF8010070 (1)[1], 0xF80100B0 (1)[2]

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	NODIVCLK
7	6	5	4	3	2	1	0
–	–	TRIGSRCB		–	–	TRIGSRCA	

- **TRIGSRCA: Trigger Source for Input A**

Value	Name	Description
0	EXTERNAL_TIOAx	The trigger/capture input A is driven by external pin TIOAx
1	PWMx	The trigger/capture input A is driven internally by PWMx

- **TRIGSRCB: Trigger Source for Input B**

Value	Name	Description
0	EXTERNAL_TIOBx	The trigger/capture input B is driven by external pin TIOBx
1	PWMx	For TC0 to TC10: The trigger/capture input B is driven internally by the comparator output (see Figure 51-16) of the PWMx. For TC11: The trigger/capture input B is driven internally by the GTSUCOMP signal of the Ethernet MAC (GMAC).

- **NODIVCLK: No Divided Clock**

0: The selected clock is defined by field TCCLKS in TC_CMRx.

1: The selected clock is peripheral clock and TCCLKS field (TC_CMRx) has no effect.

51.7.15 TC Block Control Register

Name: TC_BCR

Address: 0xF800C0C0 (0), 0xF80100C0 (1)

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	SYNC

- **SYNC: Synchro Command**

0: No effect.

1: Asserts the SYNC signal which generates a software trigger simultaneously for each of the channels.

51.7.16 TC Block Mode Register

Name: TC_BMR

Address: 0xF800C0C4 (0), 0xF80100C4 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	MAXCMP				MAXFILT	

23	22	21	20	19	18	17	16
MAXFILT				–	AUTOC	IDXPHB	SWAP

15	14	13	12	11	10	9	8
INVIDX	INVB	INVA	EDGPHA	QDTRANS	SPEEDEN	POSEN	QDEN

7	6	5	4	3	2	1	0
–	–	TC2XC2S		TC1XC1S		TC0XC0S	

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

- **TC0XC0S: External Clock Signal 0 Selection**

Value	Name	Description
0	TCLK0	Signal connected to XC0: TCLK0
1	–	Reserved
2	TIOA1	Signal connected to XC0: TIOA1
3	TIOA2	Signal connected to XC0: TIOA2

- **TC1XC1S: External Clock Signal 1 Selection**

Value	Name	Description
0	TCLK1	Signal connected to XC1: TCLK1
1	–	Reserved
2	TIOA0	Signal connected to XC1: TIOA0
3	TIOA2	Signal connected to XC1: TIOA2

- **TC2XC2S: External Clock Signal 2 Selection**

Value	Name	Description
0	TCLK2	Signal connected to XC2: TCLK2
1	–	Reserved
2	TIOA0	Signal connected to XC2: TIOA0
3	TIOA1	Signal connected to XC2: TIOA1

- **QDEN: Quadrature Decoder Enabled**

0: Disabled.

1: Enables the QDEC (filter, edge detection and quadrature decoding).

Quadrature decoding (direction change) can be disabled using QDTRANS bit.

One of the POSEN or SPEEDEN bits must be also enabled.

- **POSEN: Position Enabled**

0: Disable position.

1: Enables the position measure on channel 0 and 1.

- **SPEEDEN: Speed Enabled**

0: Disabled.

1: Enables the speed measure on channel 0, the time base being provided by channel 2.

- **QDTRANS: Quadrature Decoding Transparent**

0: Full quadrature decoding logic is active (direction change detected).

1: Quadrature decoding logic is inactive (direction change inactive) but input filtering and edge detection are performed.

- **EDGPHA: Edge on PHA Count Mode**

0: Edges are detected on PHA only.

1: Edges are detected on both PHA and PHB.

- **INVA: Inverted PHA**

0: PHA (TIOA0) is directly driving the QDEC.

1: PHA is inverted before driving the QDEC.

- **INVB: Inverted PHB**

0: PHB (TIOB0) is directly driving the QDEC.

1: PHB is inverted before driving the QDEC.

- **INVIDX: Inverted Index**

0: IDX (TIOA1) is directly driving the QDEC.

1: IDX is inverted before driving the QDEC.

- **SWAP: Swap PHA and PHB**

0: No swap between PHA and PHB.

1: Swap PHA and PHB internally, prior to driving the QDEC.

- **IDXPHB: Index Pin is PHB Pin**

0: IDX pin of the rotary sensor must drive TIOA1.

1: IDX pin of the rotary sensor must drive TIOB0.

- **AUTOC: Auto-Correction of missing pulses**

0 (DISABLED): The detection and auto-correction function is disabled.

1 (ENABLED): The detection and auto-correction function is enabled.

- **MAXFILT: Maximum Filter**

1–63: Defines the filtering capabilities.

Pulses with a period shorter than MAXFILT+1 peripheral clock cycles are discarded.

- **MAXCMP: Maximum Consecutive Missing Pulses**

0: The flag MPE in TC_QISR never rises.

1–15: Defines the number of consecutive missing pulses before a flag report.

51.7.17 TC QDEC Interrupt Enable Register

Name: TC_QIER

Address: 0xF800C0C8 (0), 0xF80100C8 (1)

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	QERR	DIRCHG	IDX

- **IDX: Index**

0: No effect.

1: Enables the interrupt when a rising edge occurs on IDX input.

- **DIRCHG: Direction Change**

0: No effect.

1: Enables the interrupt when a change on rotation direction is detected.

- **QERR: Quadrature Error**

0: No effect.

1: Enables the interrupt when a quadrature error occurs on PHA, PHB.

51.7.18 TC QDEC Interrupt Disable Register

Name: TC_QIDR

Address: 0xF800C0CC (0), 0xF80100CC (1)

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	QERR	DIRCHG	IDX

- **IDX: Index**

0: No effect.

1: Disables the interrupt when a rising edge occurs on IDX input.

- **DIRCHG: Direction Change**

0: No effect.

1: Disables the interrupt when a change on rotation direction is detected.

- **QERR: Quadrature Error**

0: No effect.

1: Disables the interrupt when a quadrature error occurs on PHA, PHB.

51.7.19 TC QDEC Interrupt Mask Register

Name: TC_QIMR

Address: 0xF800C0D0 (0), 0xF80100D0 (1)

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	QERR	DIRCHG	IDX

- **IDX: Index**

0: The interrupt on IDX input is disabled.

1: The interrupt on IDX input is enabled.

- **DIRCHG: Direction Change**

0: The interrupt on rotation direction change is disabled.

1: The interrupt on rotation direction change is enabled.

- **QERR: Quadrature Error**

0: The interrupt on quadrature error is disabled.

1: The interrupt on quadrature error is enabled.

51.7.20 TC QDEC Interrupt Status Register

Name: TC_QISR

Address: 0xF800C0D4 (0), 0xF80100D4 (1)

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	DIR
7	6	5	4	3	2	1	0
–	–	–	–	MPE	QERR	DIRCHG	IDX

- **IDX: Index**

0: No Index input change since the last read of TC_QISR.

1: The IDX input has changed since the last read of TC_QISR.

- **DIRCHG: Direction Change**

0: No change on rotation direction since the last read of TC_QISR.

1: The rotation direction changed since the last read of TC_QISR.

- **QERR: Quadrature Error**

0: No quadrature error since the last read of TC_QISR.

1: A quadrature error occurred since the last read of TC_QISR.

- **MPE: Consecutive Missing Pulse Error**

0: The number of consecutive missing pulses has not reached the maximum value specified in MAXMP since the last read of TC_QISR.

1: An occurrence of MAXCMP consecutive missing pulses has been detected since the last read of TC_QISR.

- **DIR: Direction**

Returns an image of the actual rotation direction.

51.7.21 TC Fault Mode Register

Name: TC_FMR

Address: 0xF800C0D8 (0), 0xF80100D8 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	ENCF1	ENCF0

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

- **ENCF0: Enable Compare Fault Channel 0**

0: Disables the FAULT output source (CPCS flag) from channel 0.

1: Enables the FAULT output source (CPCS flag) from channel 0.

- **ENCF1: Enable Compare Fault Channel 1**

0: Disables the FAULT output source (CPCS flag) from channel 1.

1: Enables the FAULT output source (CPCS flag) from channel 1.

51.7.22 TC Write Protection Mode Register

Name: TC_WPMR

Address: 0xF800C0E4 (0), 0xF80100E4 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
WPKEY							

23	22	21	20	19	18	17	16
WPKEY							

15	14	13	12	11	10	9	8
WPKEY							

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	WPEN

- **WPEN: Write Protection Enable**

0: Disables the write protection if WPKEY corresponds to 0x54494D (“TIM” in ASCII).

1: Enables the write protection if WPKEY corresponds to 0x54494D (“TIM” in ASCII).

The Timer Counter clock of the first channel must be enabled to access this register.

See Section 51.6.19 “Register Write Protection” for a list of registers that can be write-protected and Timer Counter clock conditions.

- **WPKEY: Write Protection Key**

Value	Name	Description
0x54494D	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

52. Pulse Density Modulation Interface Controller (PDMIC)

52.1 Description

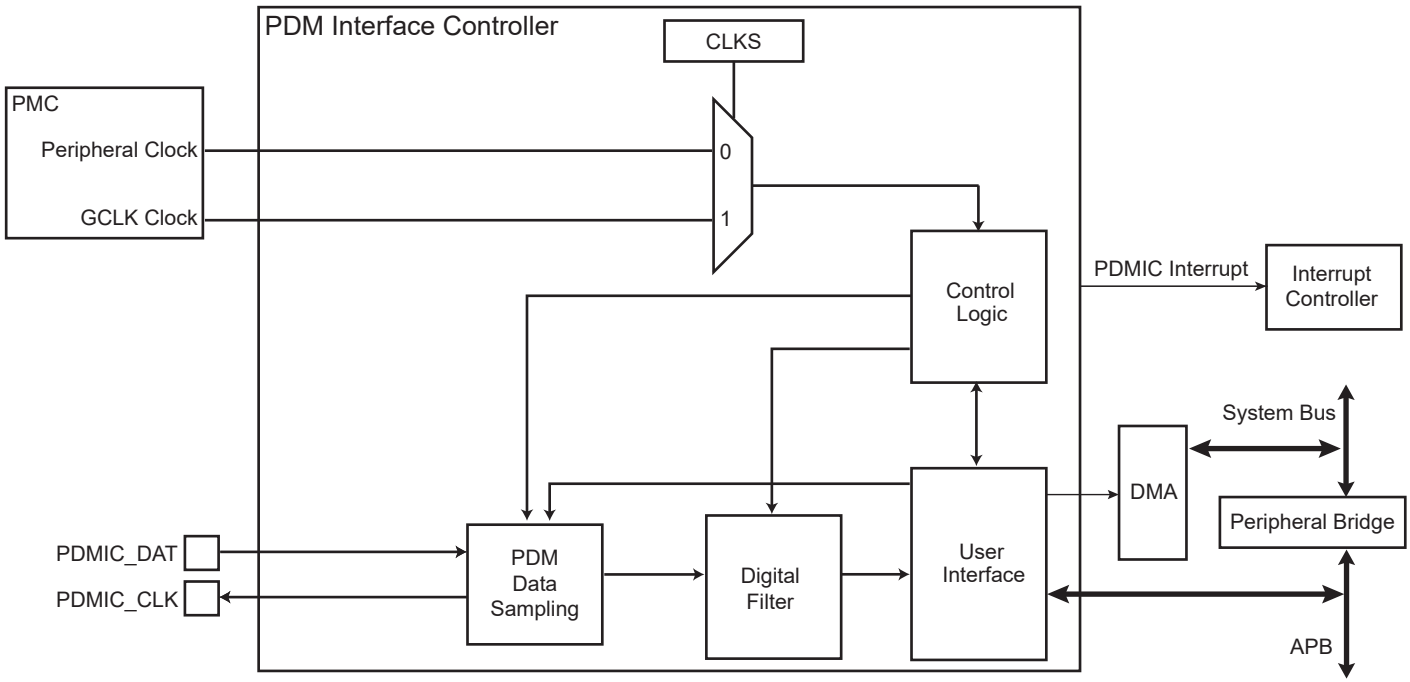
The Pulse Density Modulation Interface Controller (PDMIC) is a PDM interface controller and decoder that support mono PDM format. It integrates a clock generator driving the PDM microphone and embeds filters which decimate the incoming bitstream to obtain most common audio rates.

52.2 Embedded Characteristics

- 16-bit Resolution
- DMA Controller Support
- Up to 4 Conversions Stored
- PDM Clock Source can be Independent from Core Clock
- Register Write Protection

52.3 Block Diagram

Figure 52-1. PDMIC Block Diagram



52.4 Signal Description

Table 52-1. PDMIC Pin Description

Pin Name	Description	Type
PDMIC_CLK	Pulse Density Modulation Bitstream Sampling Clock	Output
PDMIC_DAT	Pulse Density Modulation Data	Input

52.5 Product Dependencies

52.5.1 I/O Lines

The pins used for interfacing the PDMIC are multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the peripheral functions to PDMIC pins.

Table 52-2. I/O Lines

Instance	Signal	I/O Line	Peripheral
PDMIC	PDMIC_CLK	PB12	D
PDMIC	PDMIC_CLK	PB27	D
PDMIC	PDMIC_DAT	PB11	D
PDMIC	PDMIC_DAT	PB26	D

52.5.2 Power Management

The PDMIC is not continuously clocked. The user must first enable the PDMIC peripheral clock and the PDMIC Generic Clock in the Power Management Controller (PMC) before using the controller.

52.5.3 Interrupt Sources

The PDMIC interrupt line is connected on one of the internal sources of the Interrupt Controller. Using the PDMIC interrupt requires the Interrupt Controller to be programmed first.

Table 52-3. Peripheral IDs

Instance	ID
PDMIC	48

52.6 Functional Description

52.6.1 PDM Interface

52.6.1.1 Description

The PDM clock (PDMIC_CLK) is used to sample the PDM bitstream.

The PDMIC_CLK frequency range is between peripheral clock/2 and peripheral clock/256 or between GCLK clock/2 and GCLK clock/256, depending on the selected clock source.

The GCLK clock frequency must always be at least three times lower than the peripheral clock frequency.

The field PRESCAL in the Mode Register (PDMIC_MR) must be programmed in order to provide a PDMIC_CLK frequency compliant with the microphone parameters.

52.6.1.2 Startup Sequence

To start processing the bitstream coming from the PDM interface, follow the steps below:

1. Clear all bits in the Control Register (PDMIC_CR) or compute a soft reset using the SWRST bit of PDMIC_CR.
2. Configure the PRESCAL field in PDMIC_MR according to the microphone specifications.
3. Enable the PDM mode and start the conversions using the ENPDM bit in PDMIC_CR.

52.6.2 Digital Signal Processing (Digital Filter)

52.6.2.1 Description

The PDMIC includes a DSP section containing a decimation filter, a droop compensation filter, a sixth-order low pass filter, a first-order high pass filter and an offset and gain compensation stage. A block diagram of the DSP section is represented in Figure 52-2. DSP Block Diagram.

Data processed by the filtering section are two's complement signals defined on 24 bits.

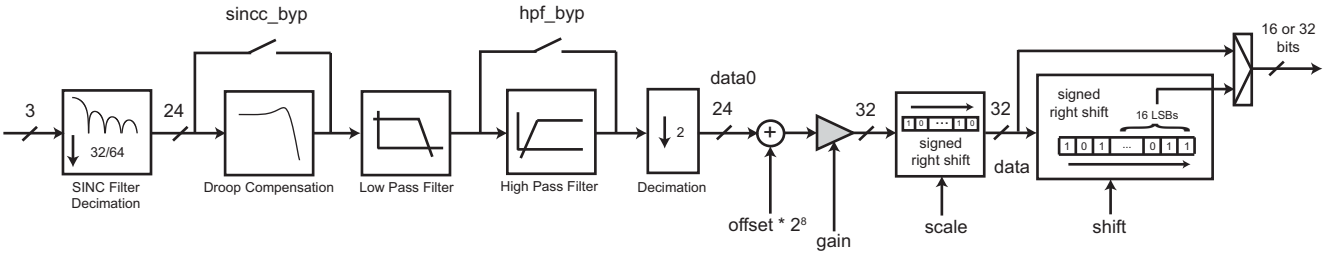
The filtering of the decimation stage is performed by a fourth-order sinc-based filter whose zeros are placed in order to minimize aliasing effects of the decimation. The decimation ratio of this filter is either 32 or 64. The droop induced by this filter can be compensated by the droop compensation stage.

The sixth-order low pass filter is used to decimate the sinc filter output by a ratio of 2.

An optional first-order high pass filter is implemented in order to eliminate the DC component of the incoming signal.

The overall decimation ratio of this DSP section is either 64 or 128. This fits an audio sampling rate of 48 kHz with a PDM microphone sampling frequency of either 3.072 or 6.144 MHz. The frequency response of the filters optimizes the gain flatness between 0 and 20 kHz (when the droop compensation filter is implemented and the high pass filter is bypassed) and highly reduces the aliasing effects of the decimation.

Figure 52-2. DSP Block Diagram



52.6.2.2 Decimation Filter

The sigma-delta architecture of the PDM microphone implies a filtering and a decimation of the bitstream at the output of the microphone bitstream. The decimation filter decimates the bitstream by either 32 or 64. To perform this operation, a fourth-order sinc filter with an Over-Sampling Ratio (OSR) of 32 or 64 is implemented with the following transfer function:

$$H(z) = \frac{1}{OSR^4}\left(\sum_{i=0}^{OSR-1} z^{-i}\right)^4$$

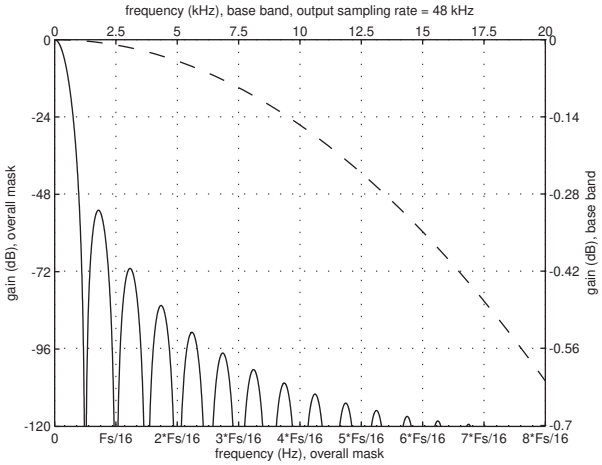
The DC gain of this filter is unity and does not depend on its OSR. However, as it generates a fourth-order zero at *Fs/OSR* frequency multiples (Fs being the sampling frequency of the microphone), the frequency response of the decimation filter depends on the OSR parameter. See Section 52.6.2.3 "Droop Compensation" for frequency plots.

Its non-flat frequency response can be compensated over the 0 to 20 kHz band by using the droop compensation filter when the decimated frequency is set to 48 kHz. See Section 52.6.2.3 "Droop Compensation".

If the decimated sampling rate is modified, the frequency response of this filter is scaled proportionally to the new frequency.

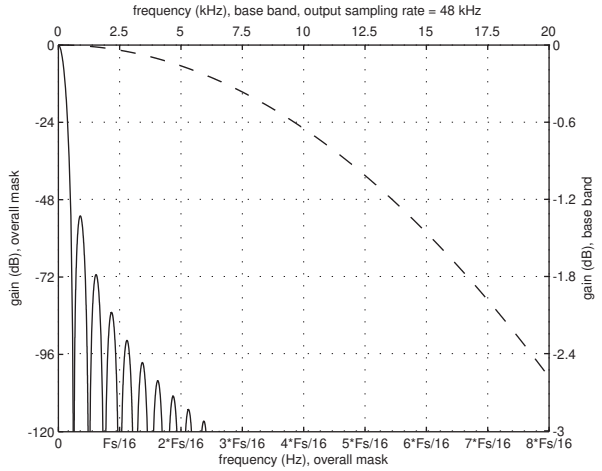
In Figure 52-3 and Figure 52-4, Fs is the sampling rate of the PDM microphone.

Figure 52-3. Spectral mask of an OSR = 32, Fs = 6.144 MHz, Fourth-Order Sinc Filter: Overall Response (continuous line) and 0 to 20 kHz Bandwidth Response (dashed line)



The zeros of this filter are located at multiples of Fs/32

Figure 52-4. Spectral Mask of an OSR = 64, Fs = 3.072 MHz, Fourth-order Sinc Filter: Overall Response (continuous line) and 0 to 20 kHz Bandwidth Response (dashed line)



The zeros of this filter are located at multiples of Fs/64.

52.6.2.3 Droop Compensation

The droop effect introduced by the sinc filter can be compensated in the 0 to 20 kHz by the droop compensation filter (see Figure 52-5). This is a second-order IIR filter which is applied on the signal output by the sinc. The default coefficients of the droop compensation filter are computed to optimize the droop of the sinc filter with the decimated frequency equal to 48 kHz.

This filter compensates the droop of the sinc filter regardless of the OSR value.

If the decimated sampling rate is modified, the frequency response of this filter is scaled proportionally to the new frequency.

This filter can be bypassed by setting the SINBYP bit in the PDMIC DSP Configuration Register 0 (PDMIC_DSPR0).

Figure 52-5. Droop Compensation Filter Overall Frequency Response

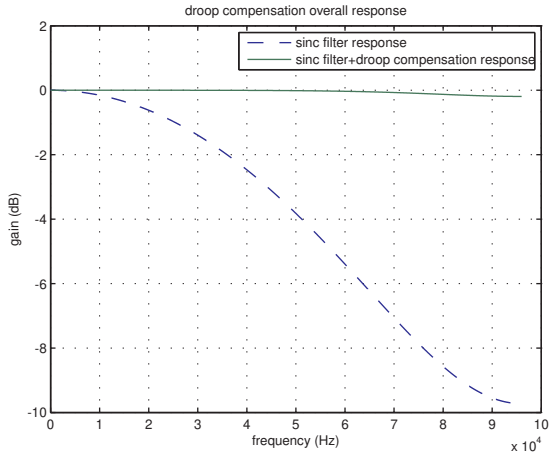
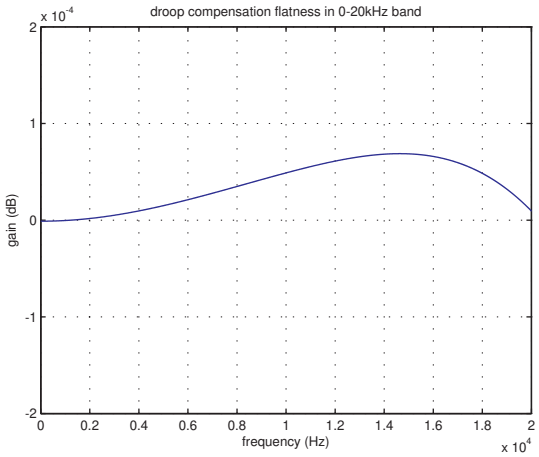


Figure 52-6. Droop Compensation Filter 0 to 20 kHz Band Flatness



52.6.2.4 Low Pass Filter

The PDMIC includes a sixth-order IIR filter that performs a low pass transfer function and decimates by 2 the output of the sinc filter. The coefficients are computed for a decimated sampling rate of 48 kHz and optimize the 0 to 20 kHz band flatness while rejecting the aliasing of the PDM microphone by at least 60 dB in the 28 to 48 kHz band.

If the decimated sampling rate is modified, the frequency response of this filter is scaled proportionally to the new frequency.

Figure 52-7 and Figure 52-8 are drawn for an output sampling frequency of 48 kHz.

Figure 52-7. Low Pass Filter Spectral Mask

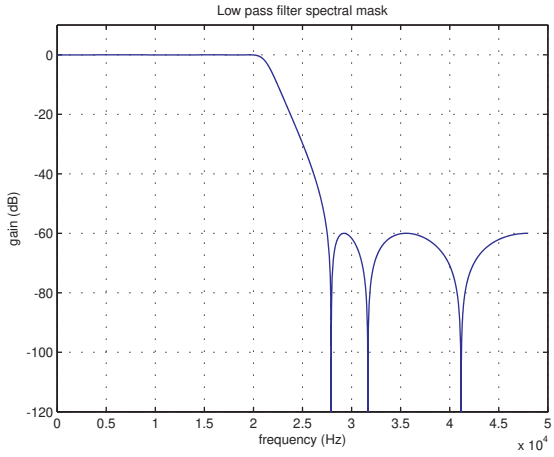
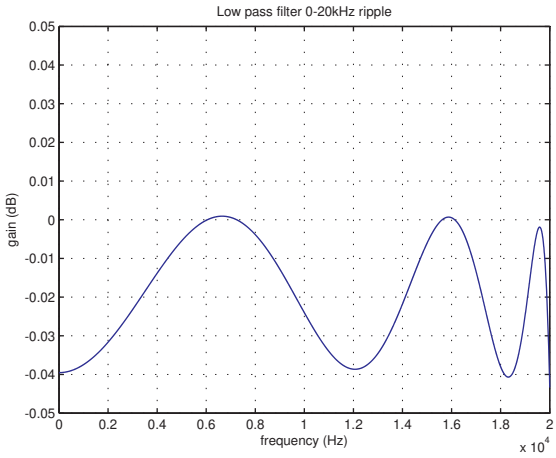


Figure 52-8. Low Pass Filter Ripple in the 0 to 20 kHz Band



52.6.2.5 High Pass Filter

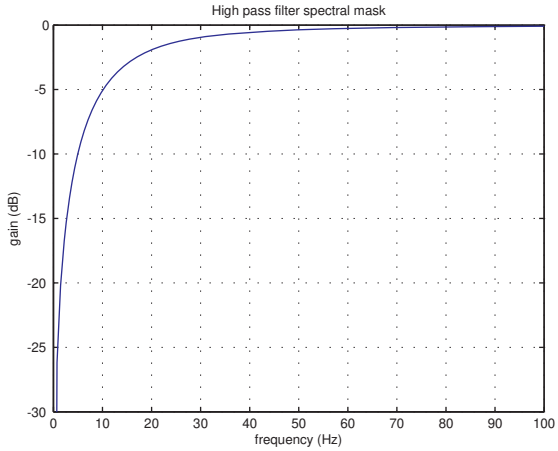
The PDMIC includes an optional first-order IIR filter performing a high pass transfer function after the low pass filter and before the decimation. The coefficients are computed for a decimated sampling rate of 48 kHz to obtain a -3dB cutoff frequency at 15 Hz.

If the decimated sampling rate is modified, the frequency response of this filter is scaled proportionally to the new frequency.

This filter can be bypassed by setting the HPFBYP bit in PDMIC_DSPR0 (see Section 52.7.8 “PDMIC DSP Configuration Register 0”).

Figure 52-9 is drawn for an output sampling frequency of 48 kHz.

Figure 52-9. High Pass Filter Spectral Mask in the 0 to 100 Hz Band



52.6.2.6 Gain and Offset Compensation

An offset, a gain, a scaling factor and a shift can be applied to a converted PDM microphone value using the following operation:

$$data = \frac{(data_0 + offset \times 2^8) \times dgain}{2^{scale + shift + 8}}$$

where:

- $data_0$ is a signed integer defined on 24 bits. It is the output of the filtering channel.
- *offset* is a signed integer defined on 16 bits (see PDMIC DSP Configuration Register 1). It is multiplied by 2^8 to have the same weight as $data_0$.
- *dgain* is an unsigned integer defined on 15 bits (see PDMIC DSP Configuration Register 1). Only the 32 MSBs of the multiplication operation are used for scaling and shifting operations. dgain defaults to 0 after reset, which forces CDR to 0. It must be programmed to a non-zero value to read non-zero data into the PDMIC_CDR register.
- *scale* is an unsigned integer defined on 4 bits (see PDMIC DSP Configuration Register 0). It shifts the multiplication operation result by *scale* bits to the right. Maximum allowed value is 15.
- *shift* is an unsigned integer defined on 4 bits (see PDMIC DSP Configuration Register 0). It shifts the multiplication operation result by *shift* bits to the right. Maximum allowed value is 15.

If the data transfer is configured in 32-bit mode (see PDMIC DSP Configuration Register 0), the 2^{shift} division is not performed and the 32-bit result of the remaining operation is sent.

If the data transfer is configured in 16-bit mode, the 2^{shift} division is performed. The result is then saturated to be within $\pm(2^{15}-1)$ and the 16 LSBs of this saturation operation are sent to the controller as the result of the PDM microphone conversion.

Default parameters are defined to output a 16-bit result whatever the data transfer configuration may be.

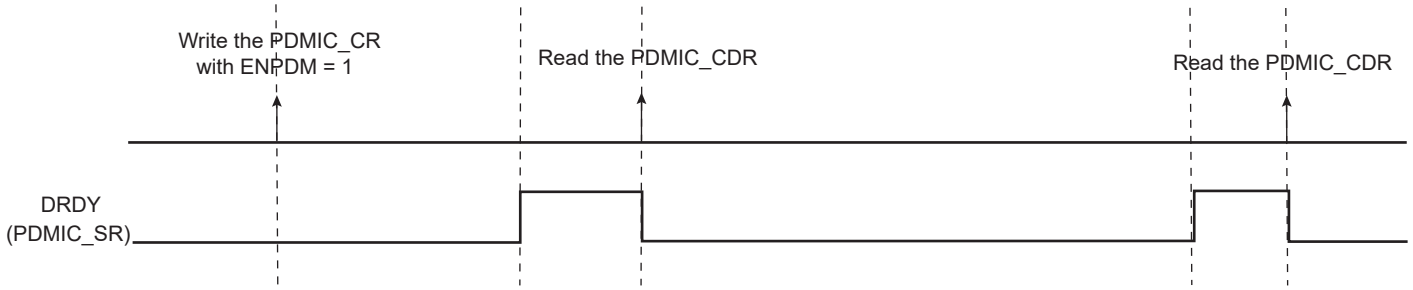
52.6.3 Conversion Results

When a conversion is completed, the resulting 16-bit digital value is stored in the PDMIC Converted Data Register (PDMIC_CDR).

The DRDY bit in the Interrupt Status Register (PDMIC_ISR) is set. In the case of a connected DMA Controller channel, DRDY rising triggers a data transfer request. In any case, DRDY can trigger an interrupt.

Reading PDMIC_CDR clears the DRDY flag.

Figure 52-10. DRDY Flag Behavior



If PDMIC_CDR is not read before further incoming data is converted, the Overrun Error (OVRE) flag is set in PDMIC_ISR. Likewise, new data converted when DRDY is high sets the OVRE bit (Overrun Error) in PDMIC_ISR. In case of overrun, the newly converted data is lost.

The OVRE flag is automatically cleared when PDMIC_ISR is read.

52.6.4 Register Write Protection

To prevent any single software error from corrupting PDMIC behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the PDMIC Write Protection Mode Register (PDMIC_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the PDMIC Write Protection Status Register (PDMIC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading PDMIC_WPSR.

The following registers can be write-protected:

- PDMIC Mode Register
- PDMIC DSP Configuration Register 0
- PDMIC DSP Configuration Register 1

52.7 Pulse Density Modulation Interface Controller (PDMIC) User Interface

Table 52-4. Register Mapping

Offset[1]	Register	Name	Access	Reset
0x00	Control Register	PDMIC_CR	Read/Write	0x00000000
0x04	Mode Register	PDMIC_MR	Read/Write	0x00F00000
0x08–0x10	Reserved	–	–	–
0x14	Converted Data Register	PDMIC_CDR	Read-only	0x00000000
0x18	Interrupt Enable Register	PDMIC_IER	Write-only	–
0x1C	Interrupt Disable Register	PDMIC_IDR	Write-only	–
0x20	Interrupt Mask Register	PDMIC_IMR	Read-only	0x00000000
0x24	Interrupt Status Register	PDMIC_ISR	Read-only	0x00000000
0x28–0x54	Reserved	–	–	–
0x58	DSP Configuration Register 0	PDMIC_DSPR0	Read/Write	0x00000000
0x5C	DSP Configuration Register 1	PDMIC_DSPR1	Read/Write	0x00000001
0x60–0xE0	Reserved	–	–	–
0xE4	Write Protection Mode Register	PDMIC_WPMR	Read/Write	0x00000000
0xE8	Write Protection Status Register	PDMIC_WPSR	Read-only	0x00000000
0xEC–0xFC	Reserved	–	–	–

Notes: 1. If an offset is not listed in the table, it must be considered as “reserved”.

52.7.1 PDMIC Control Register

Name: PDMIC_CR

Address: 0xF8018000

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	ENPDM	–	–	–	SWRST

- **SWRST: Software Reset**

0: No effect.

1: Resets the PDMIC, simulating a hardware reset.

Warning: The read value of this bit is always 0.

- **ENPDM: Enable PDM**

0: Disables the PDM and stops the conversions.

1: Enables the PDM and starts the conversions.

52.7.2 PDMIC Mode Register

Name: PDMIC_MR

Address: 0xF8018004

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	PRESCAL						
7	6	5	4	3	2	1	0
–	–	–	CLKS	–	–	–	–

This register can only be written if the WPEN bit is cleared in the PDMIC Write Protection Mode Register.

- **CLKS: Clock Source Selection**

0: Peripheral clock selected

1: GCLK clock selected (This clock source can be independent of the processor clock.)

- **PRESCAL: Prescaler Rate Selection**

PRESCAL determines the frequency of the PDM bitstream sampling clock (PDMIC_CLK):

$$PRESCAL = \frac{SELCK}{2 \times f_{PDMIC\_CLK}} - 1$$

where SELCK is either $f_{peripheral\ clock}$ or $f_{GCLK\ clock}$ depending on the value of bit CLKS ($f_{peripheral\ clock}$ or $f_{GCLK\ clock}$ is the clock frequency in Hz).

52.7.3 PDMIC Converted Data Register

Name: PDMIC_CDR

Address: 0xF8018014

Access: Read-only

31	30	29	28	27	26	25	24
DATA							
23	22	21	20	19	18	17	16
DATA							
15	14	13	12	11	10	9	8
DATA							
7	6	5	4	3	2	1	0
DATA							

- **DATA: Data Converted**

The filtered output data is placed into this register at the end of a conversion and remains until it is read.

52.7.4 PDMIC Interrupt Enable Register

Name: PDMIC_IER

Address: 0xF8018018

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	OVRE	DRDY
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

- **DRDY: Data Ready Interrupt Enable**
- **OVRE: Overrun Error Interrupt Enable**

52.7.5 PDMIC Interrupt Disable Register

Name: PDMIC_IDR

Address: 0xF801801C

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	OVRE	DRDY
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

- **DRDY: Data Ready Interrupt Disable**
- **OVRE: General Overrun Error Interrupt Disable**

52.7.6 PDMIC Interrupt Mask Register

Name: PDMIC_IMR

Address: 0xF8018020

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	OVRE	DRDY
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

- **DRDY: Data Ready Interrupt Mask**
- **OVRE: General Overrun Error Interrupt Mask**

52.7.7 PDMIC Interrupt Status Register

Name: PDMIC_ISR

Address: 0xF8018024

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	OVRE	DRDY
23	22	21	20	19	18	17	16
FIFOCNT							
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **FIFOCNT: FIFO Count**

Number of conversions available in the FIFO (not a source of interrupt).

- **DRDY: Data Ready (cleared by reading PDMIC_CDR)**

0: No data has been converted since the last read of PDMIC_CDR.

1: At least one data has been converted and is available in PDMIC_CDR.

- **OVRE: Overrun Error (cleared on read)**

0: No overrun error has occurred since the last read of PDMIC_ISR.

1: At least one overrun error has occurred since the last read of PDMIC_ISR.

52.7.8 PDMIC DSP Configuration Register 0

Name: PDMIC_DSPR0

Address: 0xF8018058

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
SHIFT				SCALE			
7	6	5	4	3	2	1	0
–	OSR			SIZE	SINBYP	HPFBYP	–

This register can only be written if the WPEN bit is cleared in the PDMIC Write Protection Mode Register.

- **HPFBYP: High-Pass Filter Bypass**

0: High-pass filter enabled.

1: Bypasses the high-pass filter.

- **SINBYP: SINCC Filter Bypass**

0: Droop compensation filter enabled.

1: Bypasses the droop compensation filter.

- **SIZE: Data Size**

0: Converted data size is 16 bits.

1: Converted data size is 32 bits.

- **OSR: Global Oversampling Ratio**

Value	Name	Description
0	128	Global Oversampling ratio is 128 (SINC filter oversampling ratio is 64)
1	64	Global Oversampling ratio is 64 (SINC filter oversampling ratio is 32)

Note: Values not listed are reserved.

- **SCALE: Data Scale**

Shifts the multiplication operation result by SCALE bits to the right.

- **SHIFT: Data Shift**

Shifts the scaled result by SHIFT bits to the right.

52.7.9 PDMIC DSP Configuration Register 1

Name: PDMIC_DSPR1

Address: 0xF801805C

Access: Read/Write

31	30	29	28	27	26	25	24
OFFSET							
23	22	21	20	19	18	17	16
OFFSET							
15	14	13	12	11	10	9	8
–	DGAIN						
7	6	5	4	3	2	1	0
DGAIN							

This register can only be written if the WPEN bit is cleared in the PDMIC Write Protection Mode Register.

- **DGAIN: Gain Correction**

Gain correction to apply to the final result.

- **OFFSET: Offset Correction**

Offset correction to apply to the final result.

DGAIN and OFFSET values can be determined using the formula in Section 52.6.2.6 “Gain and Offset Compensation”.

52.7.10 PDMIC Write Protection Mode Register

Name: PDMIC_WPMR

Address: 0xF80180E4

Access: Read/Write

31	30	29	28	27	26	25	24
WPKEY							
23	22	21	20	19	18	17	16
WPKEY							
15	14	13	12	11	10	9	8
WPKEY							
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	WPEN

- **WPEN: Write Protection Enable**

0: Disables the write protection if WPKEY corresponds to 0x414443 (“ADC” in ASCII).

1: Enables the write protection if WPKEY corresponds to 0x414443 (“ADC” in ASCII).

See Section 52.6.4 “Register Write Protection” for the list of registers that can be write-protected.

- **WPKEY: Write Protection Key**

Value	Name	Description
0x414443	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

52.7.11 PDMIC Write Protection Status Register

Name: PDMIC_WPSR

Address: 0xF80180E8

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
WPVSRC							
15	14	13	12	11	10	9	8
WPVSRC							
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	WPVS

- **WPVS: Write Protection Violation Status**

0: No write protection violation has occurred since the last read of PDMIC_WPSR.

1: A write protection violation has occurred since the last read of PDMIC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

- **WPVSRC: Write Protection Violation Source**

When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

53. Pulse Width Modulation Controller (PWM)

53.1 Description

The Pulse Width Modulation Controller (PWM) generates output pulses on 4 channels independently according to parameters defined per channel. Each channel controls two complementary square output waveforms. Characteristics of the output waveforms such as period, duty-cycle, polarity and dead-times (also called dead-bands or non-overlapping times) are configured through the user interface. Each channel selects and uses one of the clocks provided by the clock generator. The clock generator provides several clocks resulting from the division of the PWM peripheral clock. External triggers can be managed to allow output pulses to be modified in real time.

All accesses to the PWM are made through registers mapped on the peripheral bus. All channels integrate a double buffering system in order to prevent an unexpected output waveform while modifying the period, the spread spectrum, the duty-cycle or the dead-times.

Channels can be linked together as synchronous channels to be able to update their duty-cycle or dead-times at the same time.

The update of duty-cycles of synchronous channels can be performed by the DMA Controller channel which offers buffer transfer without processor Intervention.

The PWM includes a spread-spectrum counter to allow a constantly varying period (only for Channel 0). This counter may be useful to minimize electromagnetic interference or to reduce the acoustic noise of a PWM driven motor.

The PWM provides 1 independent comparison units capable of comparing a programmed value to the counter of the synchronous channels (counter of channel 0). These comparisons are intended to generate software interrupts, to trigger pulses on the 2 independent event lines (in order to synchronize ADC conversions with a lot of flexibility independently of the PWM outputs) and to trigger DMA Controller transfer requests.

PWM outputs can be overridden synchronously or asynchronously to their channel counter.

The PWM provides a fault protection mechanism with 6 fault inputs, capable to detect a fault condition and to override the PWM outputs asynchronously (outputs forced to ‘0’, ‘1’ or Hi-Z).

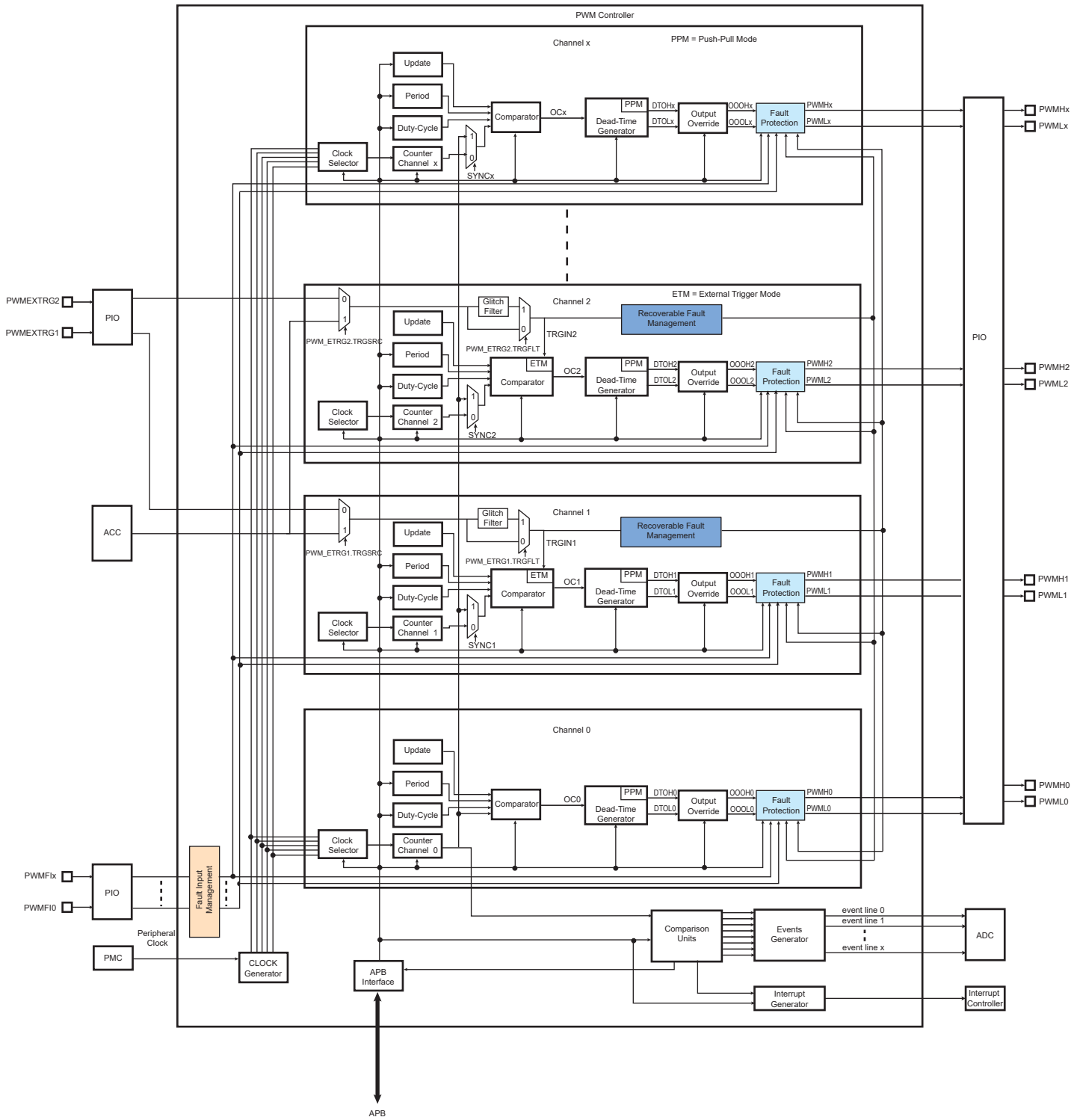
For safety usage, some configuration registers are write-protected.

53.2 Embedded Characteristics

- 4 Channels
- Common Clock Generator Providing Thirteen Different Clocks
  - A Modulo n Counter Providing Eleven Clocks
  - Two Independent Linear Dividers Working on Modulo n Counter Outputs
- Independent Channels
  - Independent 16-bit Counter for Each Channel
  - Independent Complementary Outputs with 16-bit Dead-Time Generator (Also Called Dead-Band or Non-Overlapping Time) for Each Channel
  - Independent Push-Pull Mode for Each Channel
  - Independent Enable Disable Command for Each Channel
  - Independent Clock Selection for Each Channel
  - Independent Period, Duty-Cycle and Dead-Time for Each Channel
  - Independent Double Buffering of Period, Duty-Cycle and Dead-Times for Each Channel
  - Independent Programmable Selection of The Output Waveform Polarity for Each Channel, with Double Buffering
  - Independent Programmable Center- or Left-aligned Output Waveform for Each Channel
  - Independent Output Override for Each Channel
  - Independent Interrupt for Each Channel, at Each Period for Left-Aligned or Center-Aligned Configuration
  - Independent Update Time Selection of Double Buffering Registers (Polarity, Duty Cycle) for Each Channel, at Each Period for Left-Aligned or Center-Aligned Configuration
- External Trigger Input Management (e.g., for DC/DC or Lighting Control)
  - External PWM Reset Mode
  - External PWM Start Mode
  - Cycle-By-Cycle Duty Cycle Mode
  - Leading-Edge Blanking
- Two 2-bit Gray Up/Down Channels for Stepper Motor Control
- Spread Spectrum Counter to Allow a Constantly Varying Duty Cycle (only for Channel 0)
- Synchronous Channel Mode
  - Synchronous Channels Share the Same Counter
  - Mode to Update the Synchronous Channels Registers after a Programmable Number of Periods
  - Synchronous Channels Supports Connection of one DMA Controller Channel Which Offers Buffer Transfer Without Processor Intervention To Update Duty-Cycle Registers
- 2 Independent Events Lines Intended to Synchronize ADC Conversions
  - Programmable delay for Events Lines to delay ADC measurements
- 1 Comparison Units Intended to Generate Interrupts, Pulses on Event Lines and DMA Controller Transfer Requests
- 6 Programmable Fault Inputs Providing an Asynchronous Protection of PWM Outputs
  - 2 User Driven through PIO Inputs
  - PMC Driven when Crystal Oscillator Clock Fails
  - ADC Controller Driven through Configurable Comparison Function
  - Timer/Counter Driven through Configurable Comparison Function
- Register Write Protection

53.3 Block Diagram

Figure 53-1. Pulse Width Modulation Controller Block Diagram



Note: For a more detailed illustration of the fault protection circuitry, refer to Figure 53-16 “Fault Protection”.

53.4 I/O Lines Description

Each channel outputs two complementary external I/O lines.

Table 53-1. I/O Line Description

Name	Description	Type
PWMHx	PWM Waveform Output High for channel x	Output
PWMLx	PWM Waveform Output Low for channel x	Output
PWMFIx	PWM Fault Input x	Input
PWMEXTRGy	PWM Trigger Input y	Input

53.5 Product Dependencies

53.5.1 I/O Lines

The pins used for interfacing the PWM are multiplexed with PIO lines. The programmer must first program the PIO controller to assign the desired PWM pins to their peripheral function. If I/O lines of the PWM are not used by the application, they can be used for other purposes by the PIO controller.

All of the PWM outputs may or may not be enabled. If an application requires only four channels, then only four PIO lines are assigned to PWM outputs.

Table 53-2. I/O Lines

Instance	Signal	I/O Line	Peripheral
PWM	PWMEXTRG1	PB3	D
PWM	PWMEXTRG2	PB10	C
PWM	PWMFI0	PB2	D
PWM	PWMFI1	PB9	C
PWM	PWMH0	PA30	D
PWM	PWMH1	PB0	D
PWM	PWMH2	PB5	C
PWM	PWMH3	PB7	C
PWM	PWML0	PA31	D
PWM	PWML1	PB1	D
PWM	PWML2	PB6	C
PWM	PWML3	PB8	C

53.5.2 Power Management

The PWM is not continuously clocked. The programmer must first enable the PWM clock in the Power Management Controller (PMC) before using the PWM. However, if the application does not require PWM operations, the PWM clock can be stopped when not needed and be restarted later. In this case, the PWM will resume its operations where it left off.

53.5.3 Interrupt Sources

The PWM interrupt line is connected on one of the internal sources of the Interrupt Controller. Using the PWM interrupt requires the Interrupt Controller to be programmed first.

Table 53-3. Peripheral IDs

Instance	ID
PWM	38

53.5.4 Fault Inputs

The PWM has the fault inputs connected to the different modules. Refer to the implementation of these modules within the product for detailed information about the fault generation procedure. The PWM receives faults from:

- PIO inputs
- the PMC
- the ADC controller
- Timer/Counters

Table 53-4. Fault Inputs

Fault Generator	External PWM Fault Input Number	Polarity Level(1)	Fault Input ID
PB2	PWMFI0	User-defined	0
PB9	PWMFI1	User-defined	1
PMC	–	To be configured to 1	2
ADC	–	To be configured to 1	3
Timer0	–	To be configured to 1	4
Timer1	–	To be configured to 1	5

Note: 1. FPOL field in PWMC_FMR.

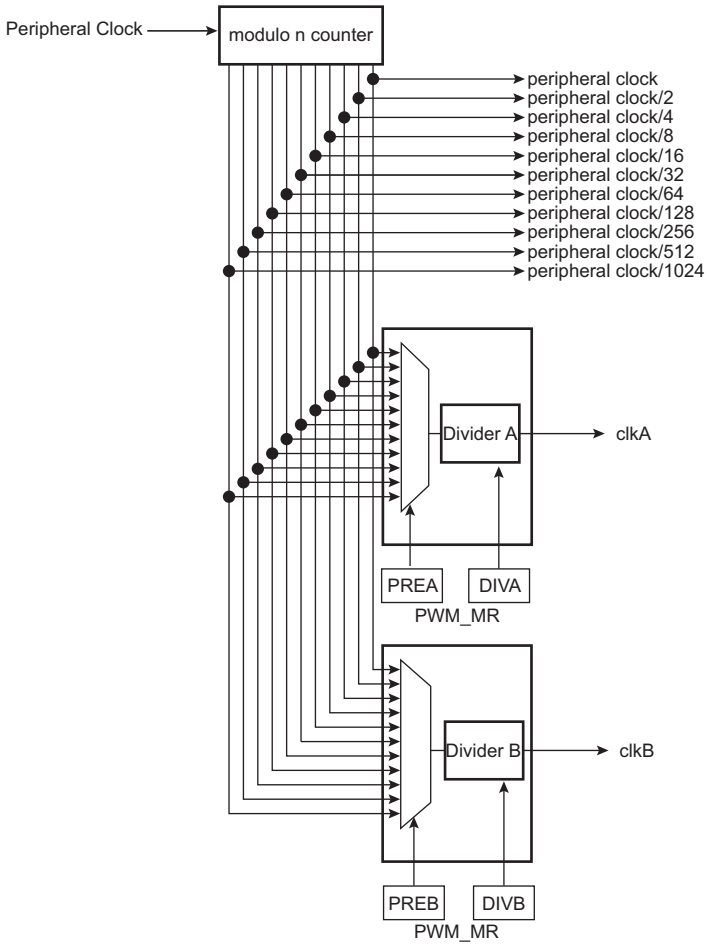
53.6 Functional Description

The PWM controller is primarily composed of a clock generator module and 4 channels.

- Clocked by the peripheral clock, the clock generator module provides 13 clocks.
- Each channel can independently choose one of the clock generator outputs.
- Each channel generates an output waveform with attributes that can be defined independently for each channel through the user interface registers.

53.6.1 PWM Clock Generator

Figure 53-2. Functional View of the Clock Generator Block Diagram



The PWM peripheral clock is divided in the clock generator module to provide different clocks available for all channels. Each channel can independently select one of the divided clocks.

The clock generator is divided into different blocks:

- a modulo n counter which provides 11 clocks: $f_{peripheral\ clock}$, $f_{peripheral\ clock}/2$, $f_{peripheral\ clock}/4$, $f_{peripheral\ clock}/8$, $f_{peripheral\ clock}/16$, $f_{peripheral\ clock}/32$, $f_{peripheral\ clock}/64$, $f_{peripheral\ clock}/128$, $f_{peripheral\ clock}/256$, $f_{peripheral\ clock}/512$, $f_{peripheral\ clock}/1024$
- two linear dividers (1, 1/2, 1/3, ... 1/255) that provide two separate clocks: clkA and clkB

Each linear divider can independently divide one of the clocks of the modulo n counter. The selection of the clock to be divided is made according to the PREA (PREB) field of the PWM Clock register (PWM_CLK). The resulting clock clkA (clkB) is the clock selected divided by DIVA (DIVB) field value.

After a reset of the PWM controller, DIVA (DIVB) and PREA (PREB) are set to ‘0’. This implies that after reset clkA (clkB) are turned off.

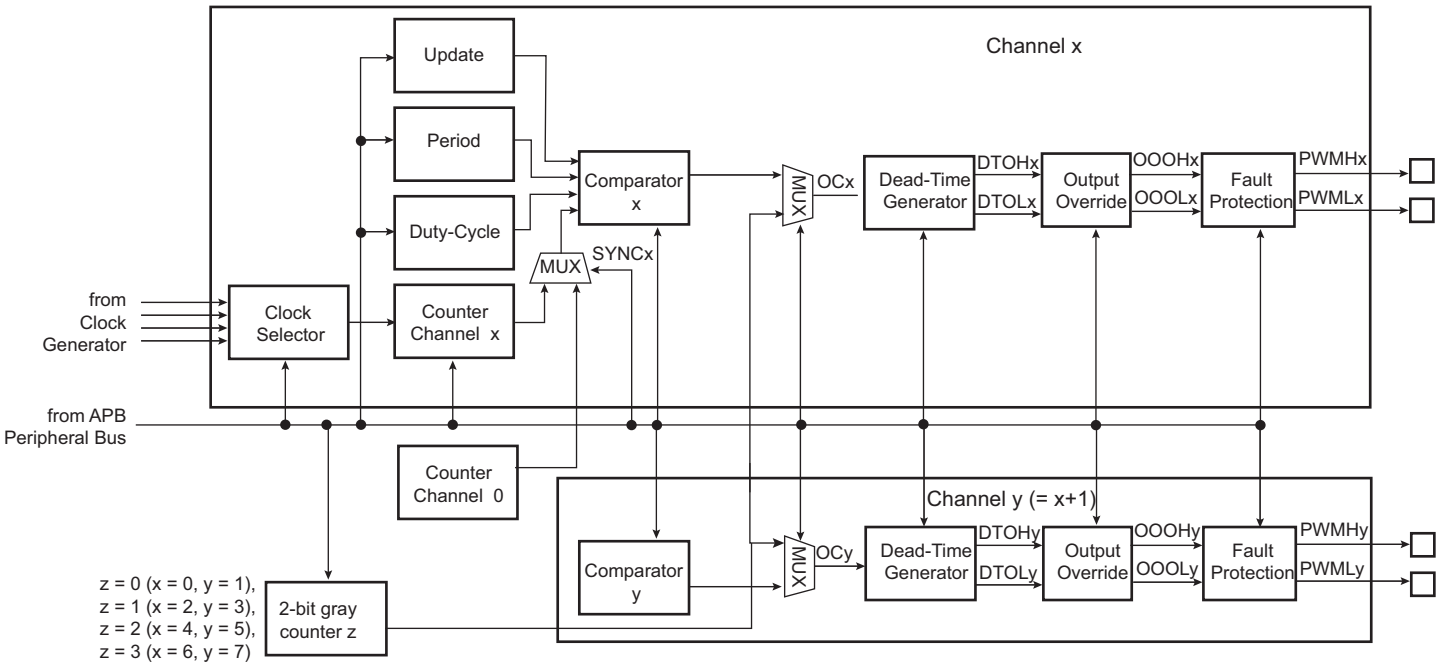
At reset, all clocks provided by the modulo n counter are turned off except the peripheral clock. This situation is also true when the PWM peripheral clock is turned off through the Power Management Controller.

CAUTION: Before using the PWM controller, the programmer must first enable the peripheral clock in the Power Management Controller (PMC).

53.6.2 PWM Channel

53.6.2.1 Channel Block Diagram

Figure 53-3. Functional View of the Channel Block Diagram



Each of the 4 channels is composed of six blocks:

- A clock selector which selects one of the clocks provided by the clock generator (described in Section 53.6.1 “PWM Clock Generator”).
- A counter clocked by the output of the clock selector. This counter is incremented or decremented according to the channel configuration and comparators matches. The size of the counter is 16 bits.
- A comparator used to compute the OCx output waveform according to the counter value and the configuration. The counter value can be the one of the channel counter or the one of the channel 0 counter according to SYNCx bit in the PWM Sync Channels Mode Register (PWM_SCM).
- A 2-bit configurable gray counter enables the stepper motor driver. One gray counter drives 2 channels.
- A dead-time generator providing two complementary outputs (DTOHx/DTOLx) which allows to drive external power control switches safely.
- An output override block that can force the two complementary outputs to a programmed value (OOOHx/OOOLx).
- An asynchronous fault protection mechanism that has the highest priority to override the two complementary outputs (PWMHx/PWMLx) in case of fault detection (outputs forced to ‘0’, ‘1’ or Hi-Z).

53.6.2.2 Comparator

The comparator continuously compares its counter value with the channel period defined by CPRD in the PWM Channel Period Register (PWM_CPRDx) and the duty-cycle defined by CDTY in the PWM Channel Duty Cycle Register (PWM_CDTYx) to generate an output signal OCx accordingly.

The different properties of the waveform of the output OCx are:

- the ***clock selection***. The channel counter is clocked by one of the clocks provided by the clock generator described in the previous section. This channel parameter is defined in the CPRE field of the PWM Channel Mode Register (PWM_CMRx). This field is reset at '0'.
- the ***waveform period***. This channel parameter is defined in the CPRD field of the PWM_CPRDx register.
  If the waveform is left-aligned, then the output waveform period depends on the counter source clock and can be calculated:
  By using the PWM peripheral clock divided by a given prescaler value "X" (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

  $$\frac{(X \times CPRD)}{f_{\text{peripheral clock}}}$$

  By using the PWM peripheral clock divided by a given prescaler value "X" (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

  $$\frac{(X \times CRPD \times DIVA)}{f_{\text{peripheral clock}}} \; or \; \frac{(X \times CRPD \times DIVB)}{f_{\text{peripheral clock}}}$$

  If the waveform is center-aligned, then the output waveform period depends on the counter source clock and can be calculated:
  By using the PWM peripheral clock divided by a given prescaler value "X" (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

  $$\frac{(2 \times X \times CPRD)}{f_{\text{peripheral clock}}}$$

  By using the PWM peripheral clock divided by a given prescaler value "X" (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

  $$\frac{(2 \times X \times CPRD \times DIVA)}{f_{\text{peripheral clock}}} \; or$$
  $$\frac{(2 \times X \times CPRD \times DIVB)}{f_{\text{peripheral clock}}}$$

- the ***waveform duty-cycle***. This channel parameter is defined in the CDTY field of the PWM_CDTYx register.
  If the waveform is left-aligned, then:

  $$\text{duty cycle} = \frac{(period - 1/\text{fchannel\_x\_clock} \times CDTY)}{period}$$

  If the waveform is center-aligned, then:

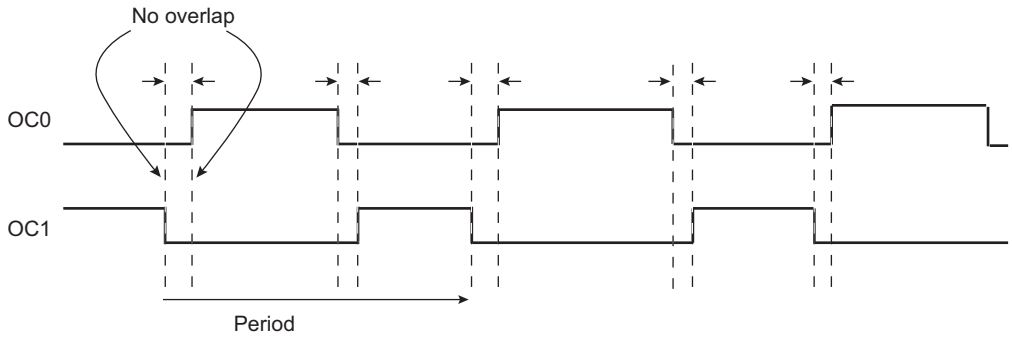
  $$\text{duty cycle} = \frac{((period/2) - 1/\text{fchannel\_x\_clock} \times CDTY))}{(period/2)}$$

- the ***waveform polarity.*** At the beginning of the period, the signal can be at high or low level. This property is defined in the CPOL bit of PWM_CMRx. By default, the signal starts by a low level. The DPOLI bit in PWM_CMRx defines the PWM polarity when the channel is disabled (CHIDx = 0 in PWM_SR). For more details, see Figure 53-5.

- DPOLI = 0: PWM polarity when the channel is disabled is the same as the one defined for the beginning of the PWM period.
- DPOLI = 1: PWM polarity when the channel is disabled is inverted compared to the one defined for the beginning of the PWM period.

- the ***waveform alignment***. The output waveform can be left- or center-aligned. Center-aligned waveforms can be used to generate non-overlapped waveforms. This property is defined in the CALG bit of PWM_CMRx. The default mode is left-aligned.

Figure 53-4. Non-Overlapped Center-Aligned Waveforms



Note: See Figure 53-5 for a detailed description of center-aligned waveforms.

When center-aligned, the channel counter increases up to CPRD and decreases down to 0. This ends the period.

When left-aligned, the channel counter increases up to CPRD and is reset. This ends the period.

Thus, for the same CPRD value, the period for a center-aligned channel is twice the period for a left-aligned channel.

Waveforms are fixed at 0 when:

- CDTY = CPRD and CPOL = 0 (Note that if TRGMODE = MODE3, the PWM waveform switches to 1 at the external trigger event (see Section 53.6.5.3 “Cycle-By-Cycle Duty Mode”)).
- CDTY = 0 and CPOL = 1

Waveforms are fixed at 1 (once the channel is enabled) when:

- CDTY = 0 and CPOL = 0
- CDTY = CPRD and CPOL = 1 (Note that if TRGMODE = MODE3, the PWM waveform switches to 0 at the external trigger event (see Section 53.6.5.3 “Cycle-By-Cycle Duty Mode”)).

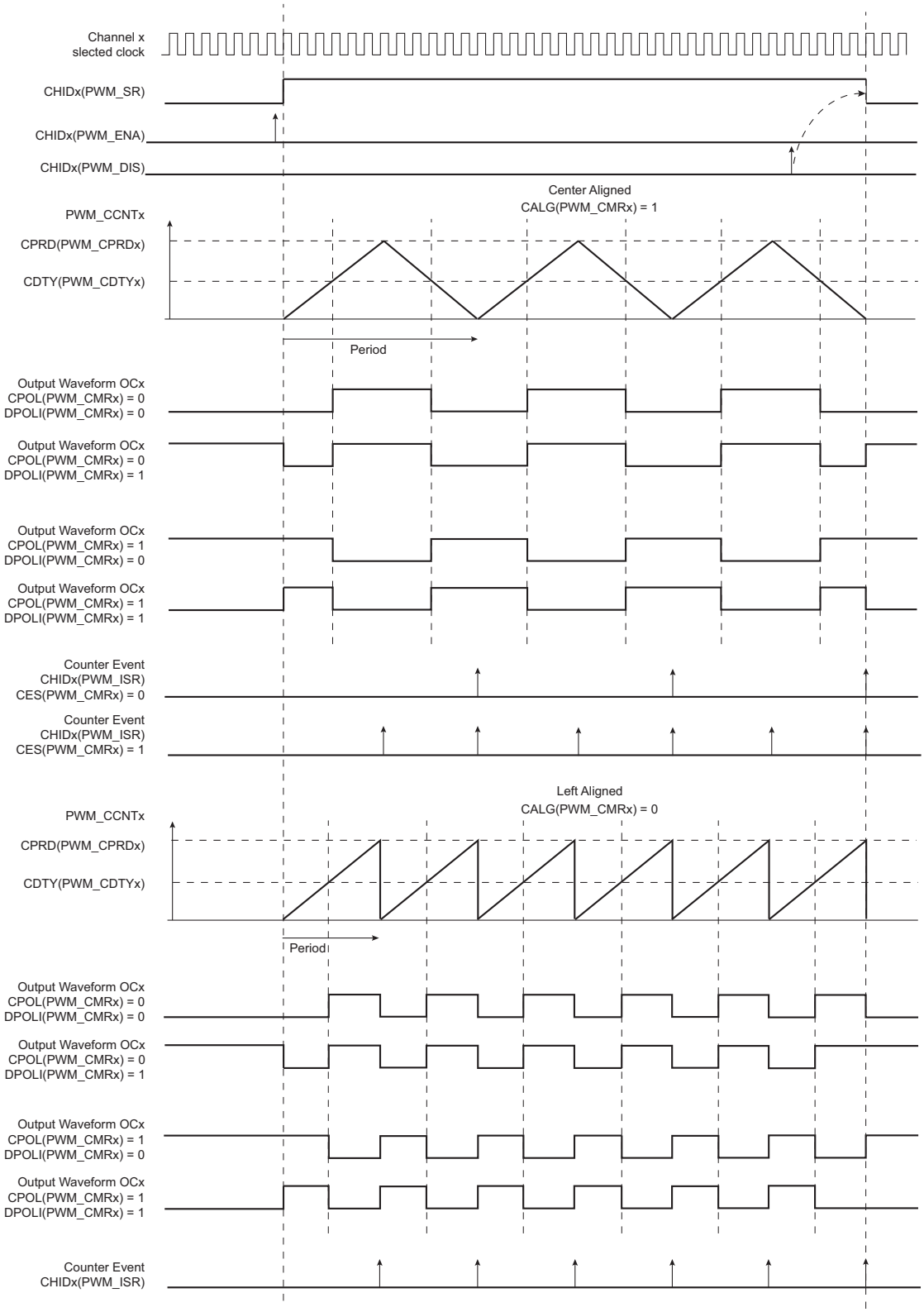
The waveform polarity must be set before enabling the channel. This immediately affects the channel output level.

Modifying CPOL in PWM Channel Mode Register while the channel is enabled can lead to an unexpected behavior of the device being driven by PWM.

In addition to generating the output signals OCx, the comparator generates interrupts depending on the counter value. When the output waveform is left-aligned, the interrupt occurs at the end of the counter period. When the output waveform is center-aligned, the bit CES of PWM_CMRx defines when the channel counter interrupt occurs. If CES is set to ‘0’, the interrupt occurs at the end of the counter period. If CES is set to ‘1’, the interrupt occurs at the end of the counter period and at half of the counter period.

Figure 53-5 illustrates the counter interrupts depending on the configuration.

Figure 53-5. Waveform Properties



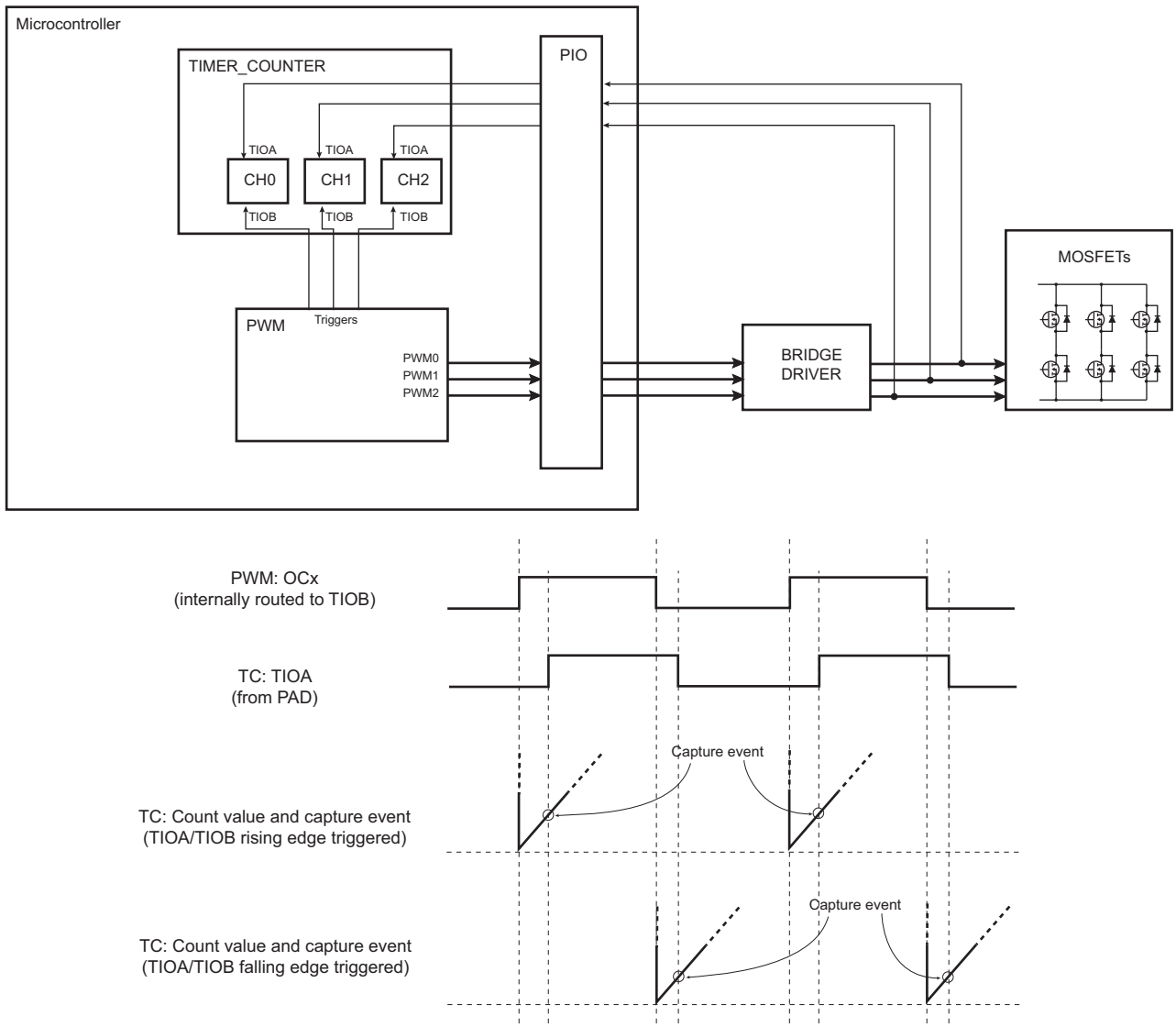
53.6.2.3 Trigger Selection for Timer Counter

The PWM controller can be used as a trigger source for the Timer Counter (TC) to achieve the two application examples described below.

Delay Measurement

To measure the delay between the channel x comparator output (OCx) and the feedback from the bridge driver of the MOSFETs (see Figure 53-6), the bit TCTS in the PWM Channel Mode Register must be at 0. This defines the comparator output of the channel x as the TC trigger source. The TIOB trigger (TC internal input) is used to start the TC; the TIOA input (from PAD) is used to capture the delay.

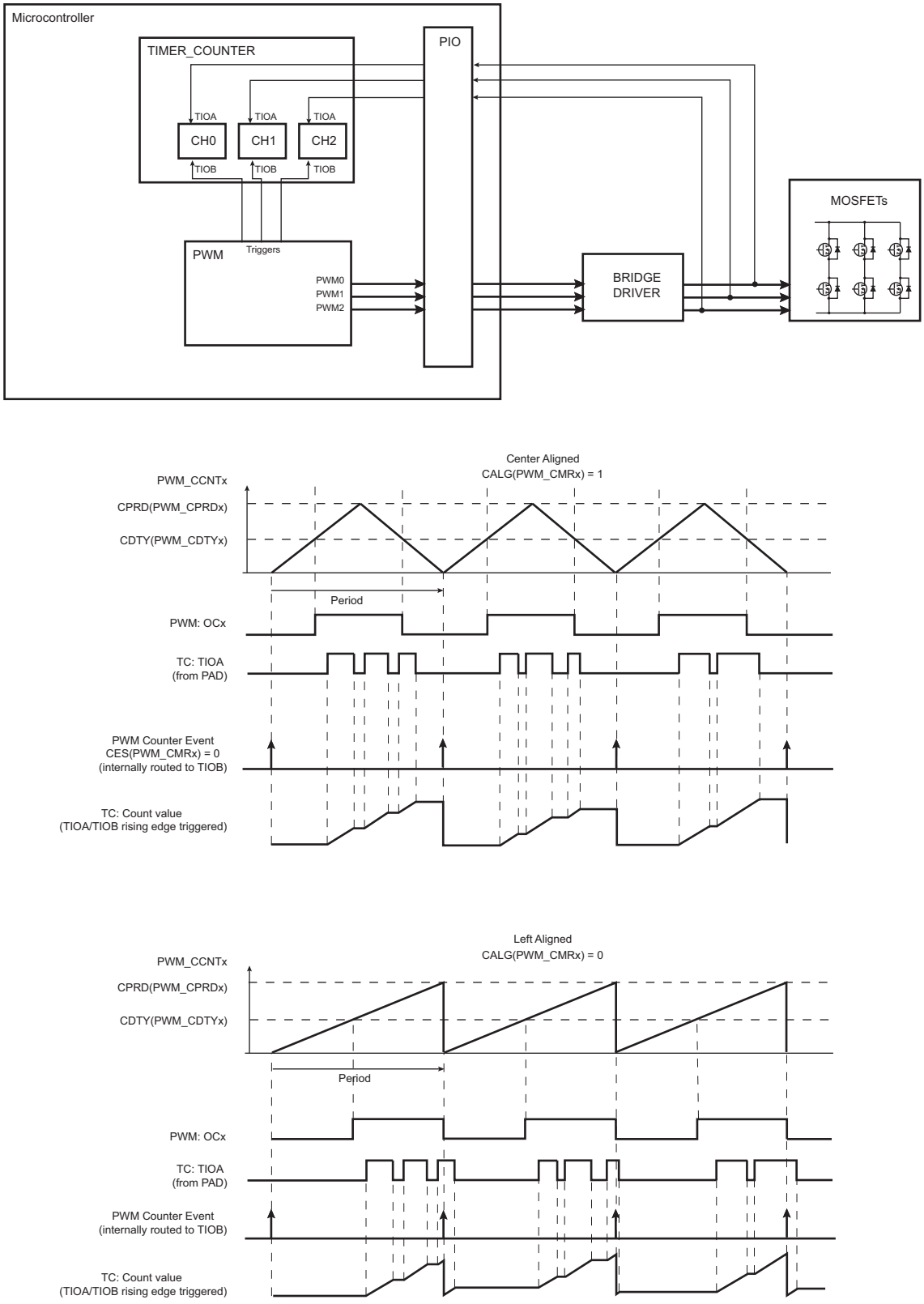
Figure 53-6. Triggering the TC: Delay Measurement



Cumulated ON Time Measurement

To measure the cumulated “ON” time of MOSFETs (see Figure 53-7), the bit TCTS of the PWM Channel Mode Register must be set to 1 to define the counter event (see Figure 53-5) as the Timer Counter trigger source.

Figure 53-7. Triggering the TC: Cumulated “ON” Time Measurement



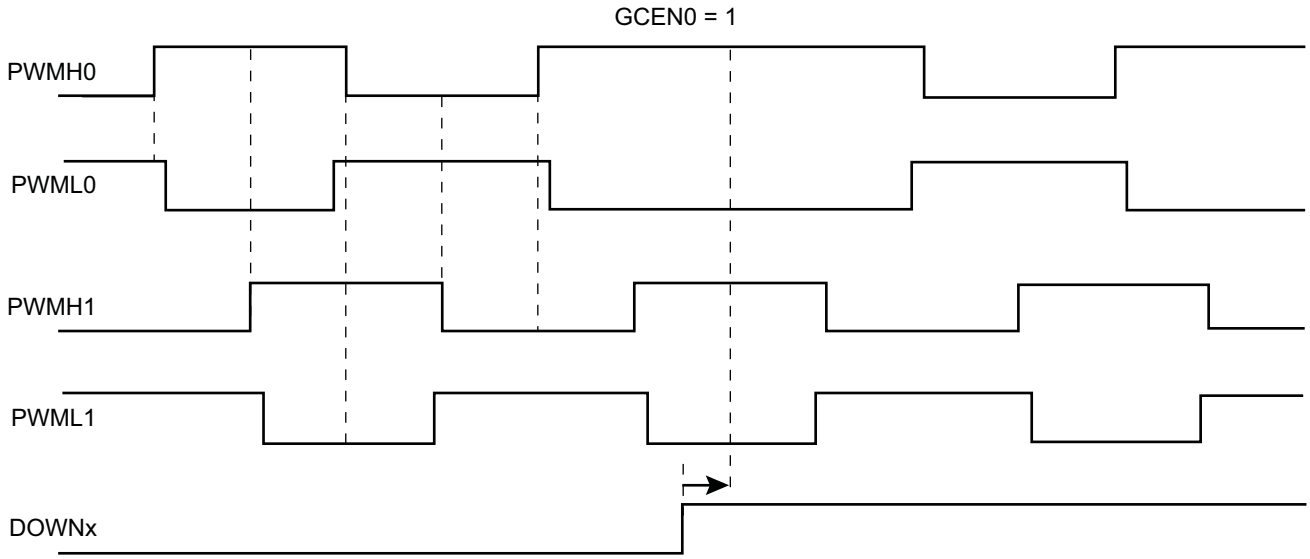
53.6.2.4 2-bit Gray Up/Down Counter for Stepper Motor

A pair of channels may provide a 2-bit gray count waveform on two outputs. Dead-time generator and other downstream logic can be configured on these channels.

Up or Down Count mode can be configured on-the-fly by means of PWM_SMMR configuration registers.

When GCEN0 is set to ‘1’, channels 0 and 1 outputs are driven with gray counter.

Figure 53-8. 2-bit Gray Up/Down Counter



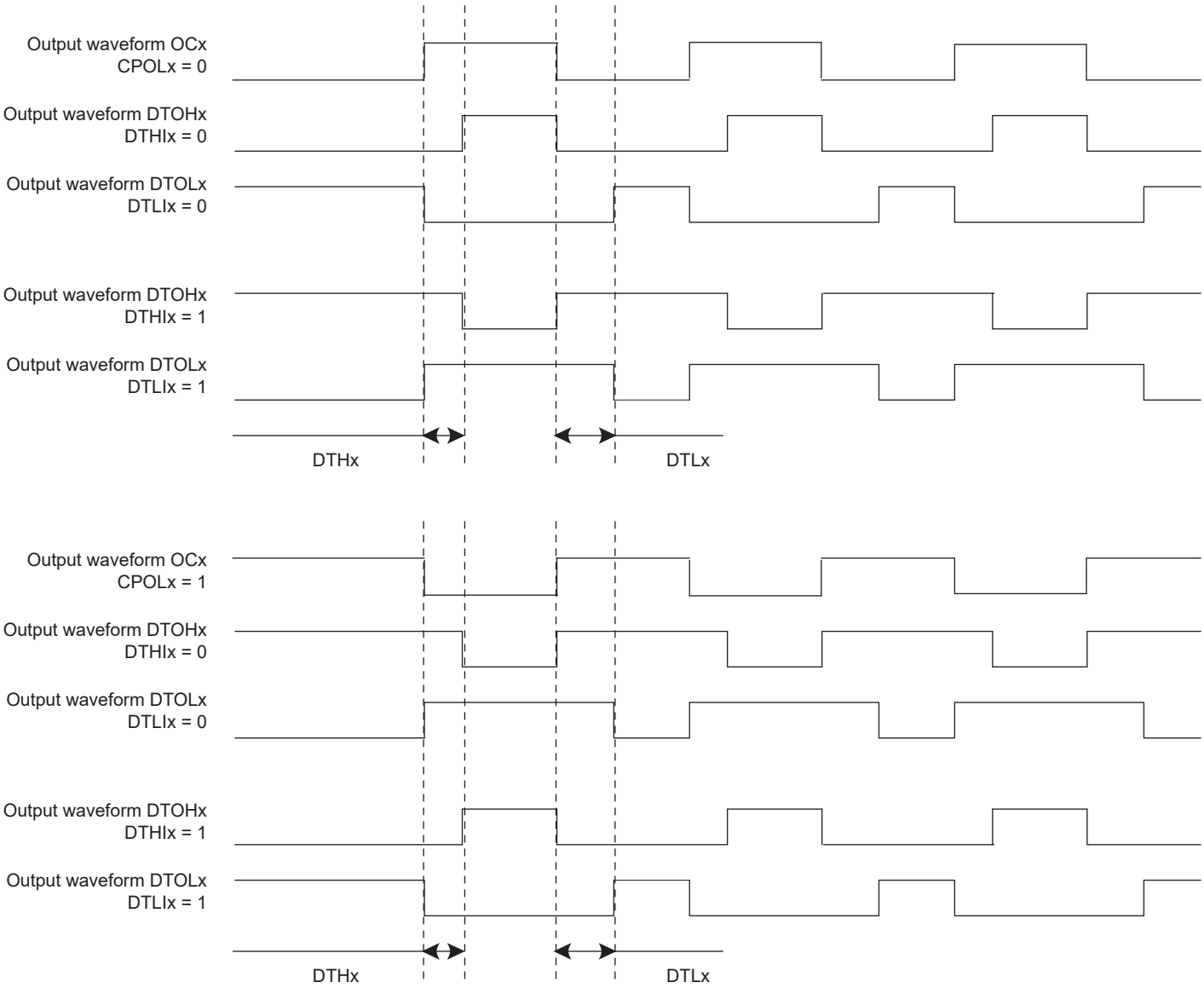
53.6.2.5 Dead-Time Generator

The dead-time generator uses the comparator output OCx to provide the two complementary outputs DTOHx and DTOLx, which allows the PWM macrocell to drive external power control switches safely. When the dead-time generator is enabled by setting the bit DTE to 1 or 0 in the PWM Channel Mode Register (PWM_CMRx), dead-times (also called dead-bands or non-overlapping times) are inserted between the edges of the two complementary outputs DTOHx and DTOLx. Note that enabling or disabling the dead-time generator is allowed only if the channel is disabled.

The dead-time is adjustable by the PWM Channel Dead Time Register (PWM_DTx). Each output of the dead-time generator can be adjusted separately by DTH and DTL. The dead-time values can be updated synchronously to the PWM period by using the PWM Channel Dead Time Update Register (PWM_DTUPDx).

The dead-time is based on a specific counter which uses the same selected clock that feeds the channel counter of the comparator. Depending on the edge and the configuration of the dead-time, DTOHx and DTOLx are delayed until the counter has reached the value defined by DTH or DTL. An inverted configuration bit (DTHI and DTLI bit in PWM_CMRx) is provided for each output to invert the dead-time outputs. The following figure shows the waveform of the dead-time generator.

Figure 53-9. Complementary Output Waveforms



PWM Push-Pull Mode

When a PWM channel is configured in Push-Pull mode, the dead-time generator output is managed alternately on each PWM cycle. The polarity of the PWM line during the idle state of the Push-Pull mode is defined by the DPOLI bit in the PWM Channel Mode Register (PWM_CMRx). The Push-Pull mode can be enabled separately on each channel by writing a one to bit PPM in the PWM Channel Mode Register.

Figure 53-10. PWM Push-Pull Mode

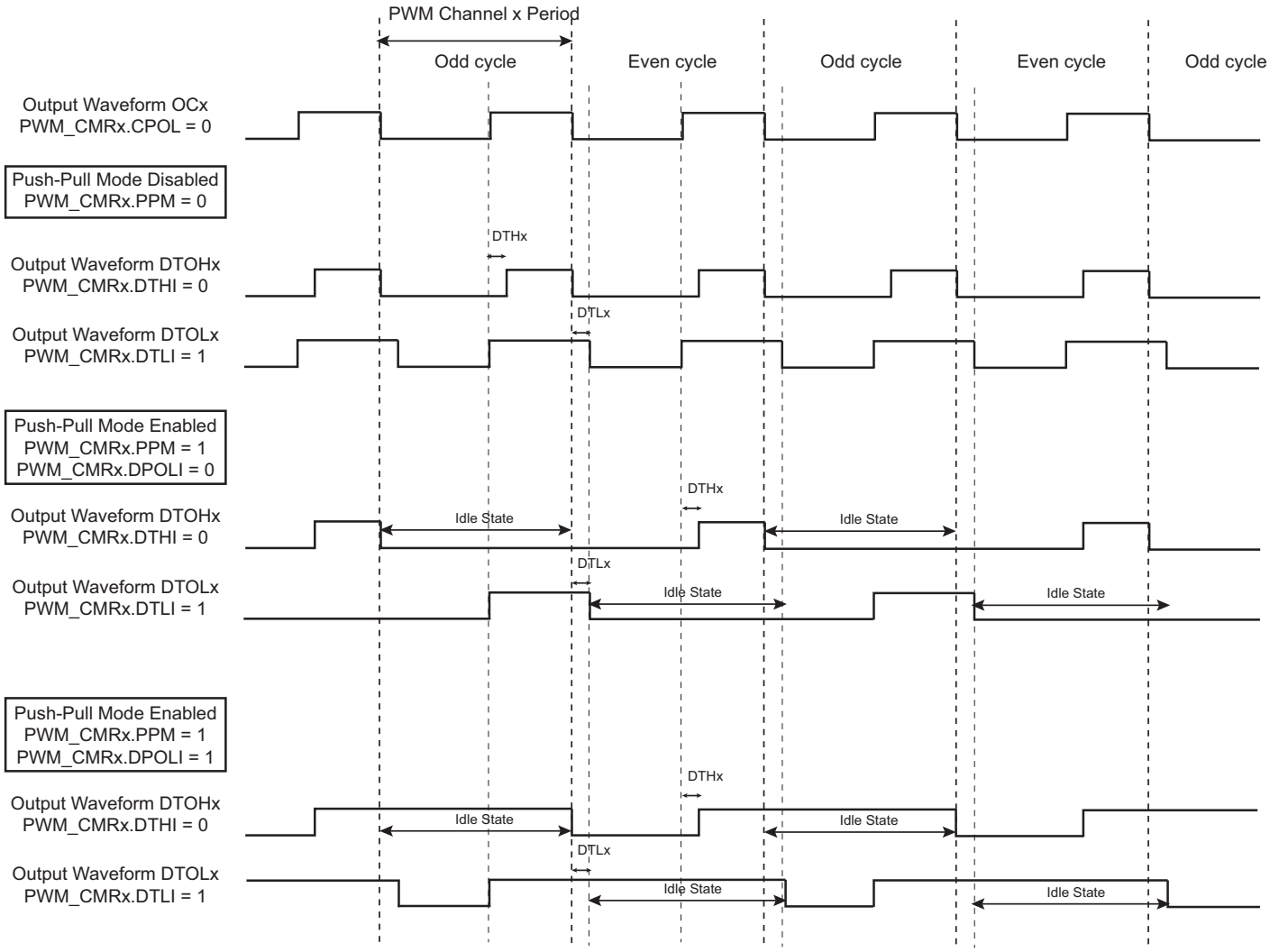


Figure 53-11. PWM Push-Pull Waveforms: Left-Aligned Mode

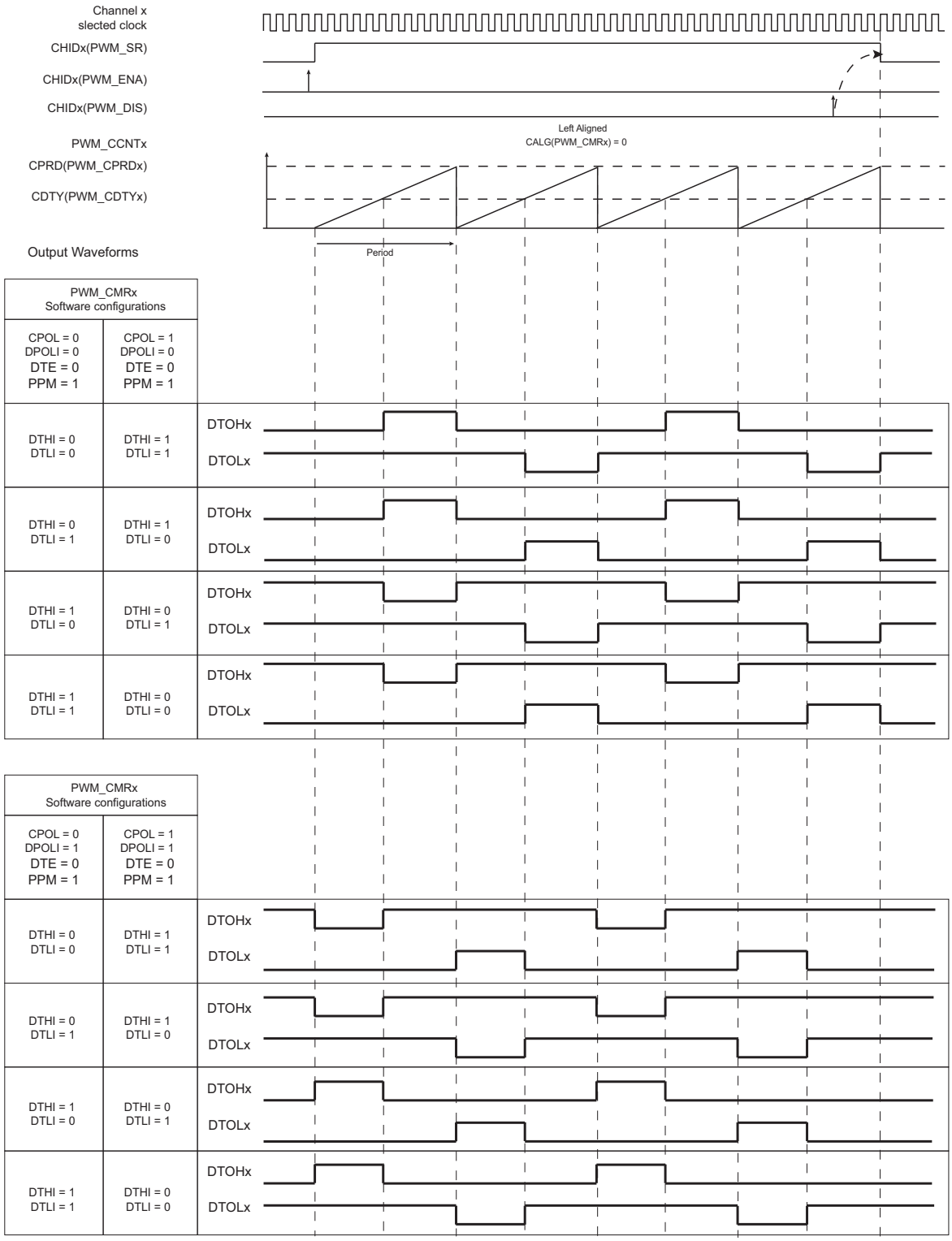
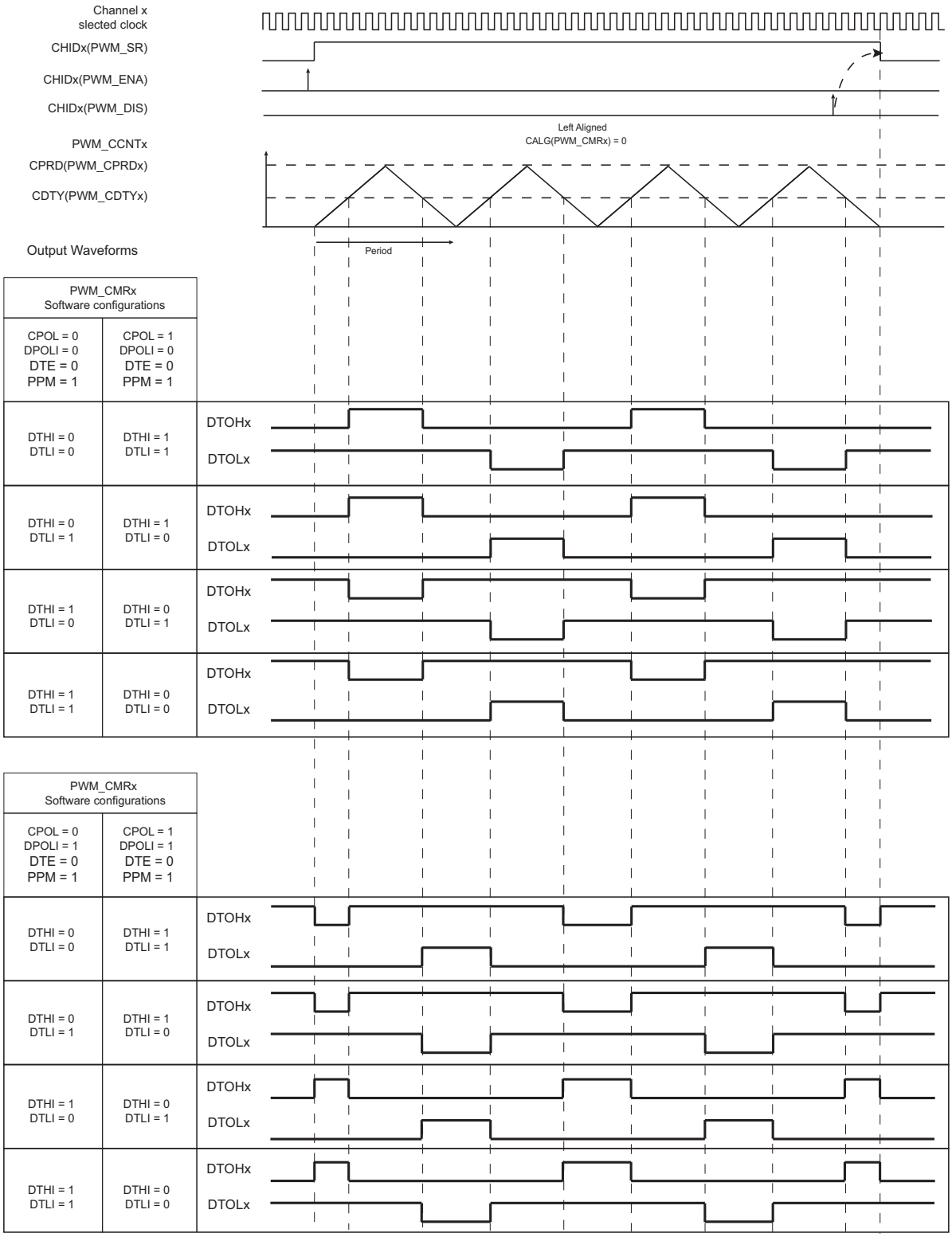


Figure 53-12. PWM Push-Pull Waveforms: Center-Aligned Mode



The PWM Push-Pull mode can be useful in transformer-based power converters, such as a half-bridge converter. The Push-Pull mode prevents the transformer core from being saturated by any direct current.

Figure 53-13. Half-Bridge Converter Application: No Feedback Regulation

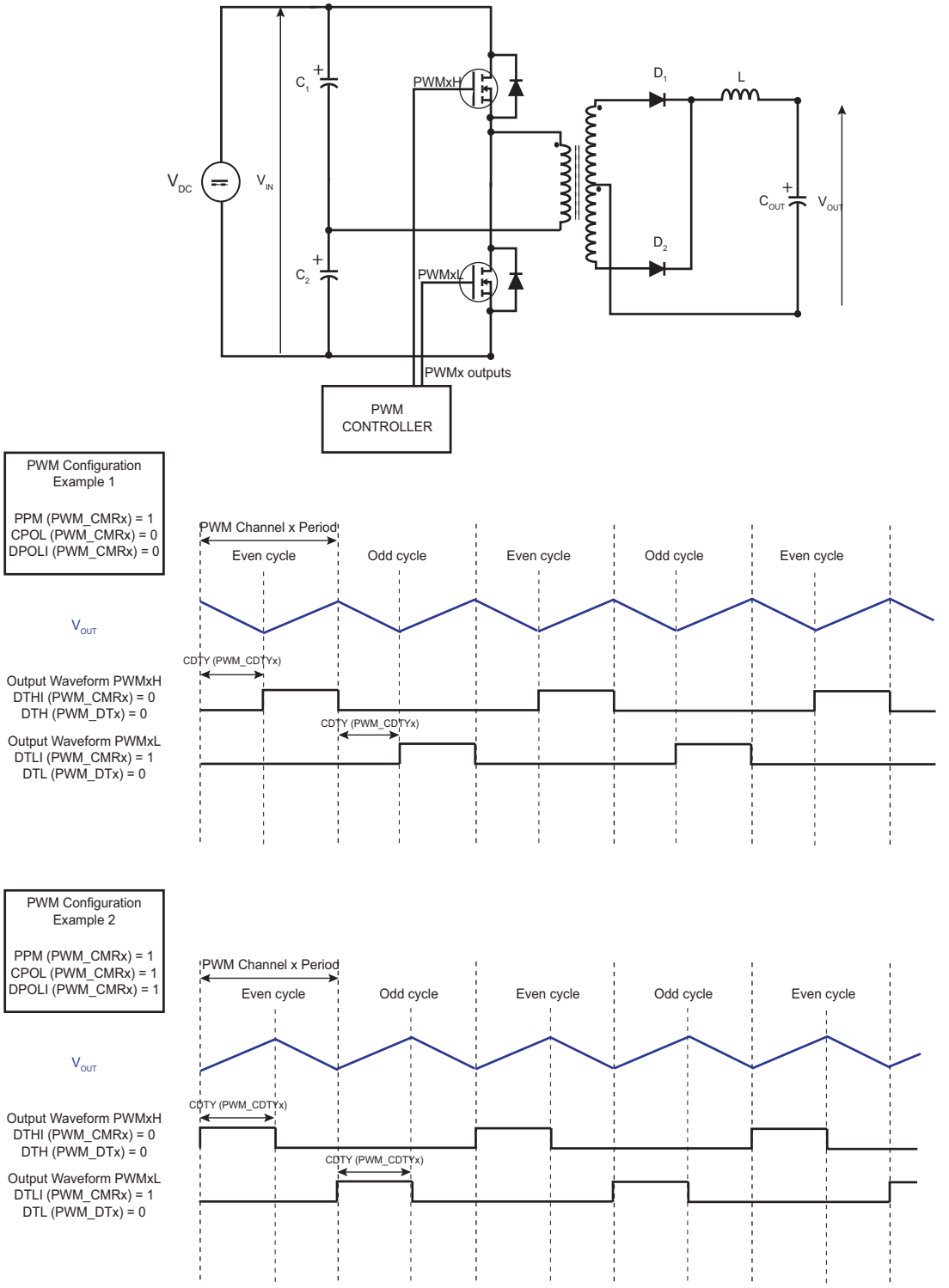
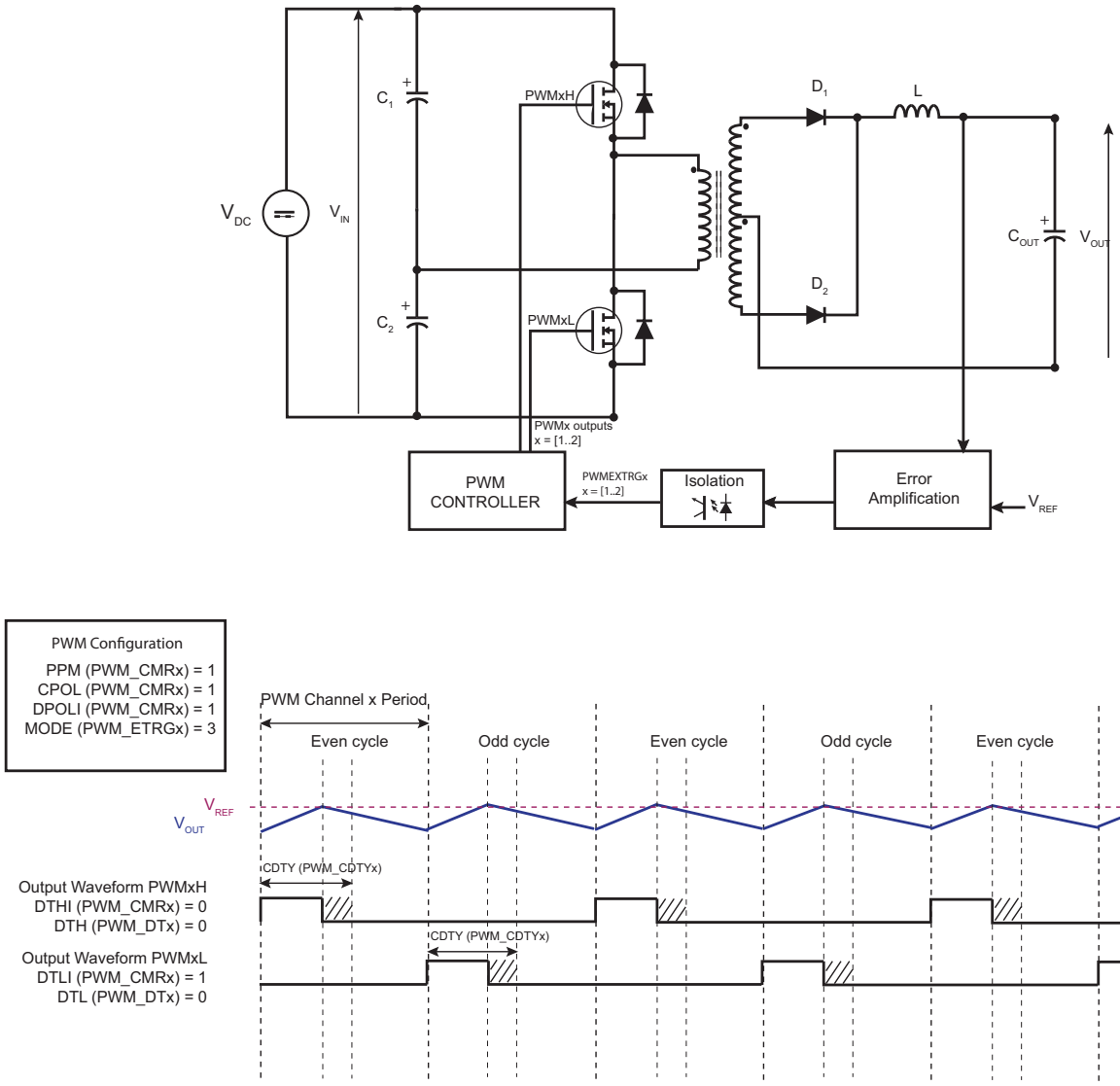


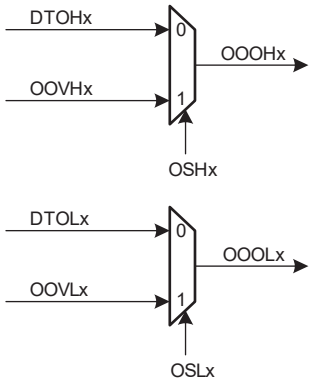
Figure 53-14. Half-Bridge Converter Application: Feedback Regulation



53.6.2.6 Output Override

The two complementary outputs DTOHx and DTOLx of the dead-time generator can be forced to a value defined by the software.

Figure 53-15. Override Output Selection



The fields OSHx and OSLx in the PWM Output Selection Register (PWM_OS) allow the outputs of the dead-time generator DTOHx and DTOLx to be overridden by the value defined in the fields OOVHx and OOVLx in the PWM Output Override Value Register (PWM_OOV).

The set registers PWM Output Selection Set Register (PWM_OSS) and PWM Output Selection Set Update Register (PWM_OSSUPD) enable the override of the outputs of a channel regardless of other channels. In the same way, the clear registers PWM Output Selection Clear Register (PWM_OSC) and PWM Output Selection Clear Update Register (PWM_OSCUPD) disable the override of the outputs of a channel regardless of other channels.

By using buffer registers PWM_OSSUPD and PWM_OSCUPD, the output selection of PWM outputs is done synchronously to the channel counter, at the beginning of the next PWM period.

By using registers PWM_OSS and PWM_OSC, the output selection of PWM outputs is done asynchronously to the channel counter, as soon as the register is written.

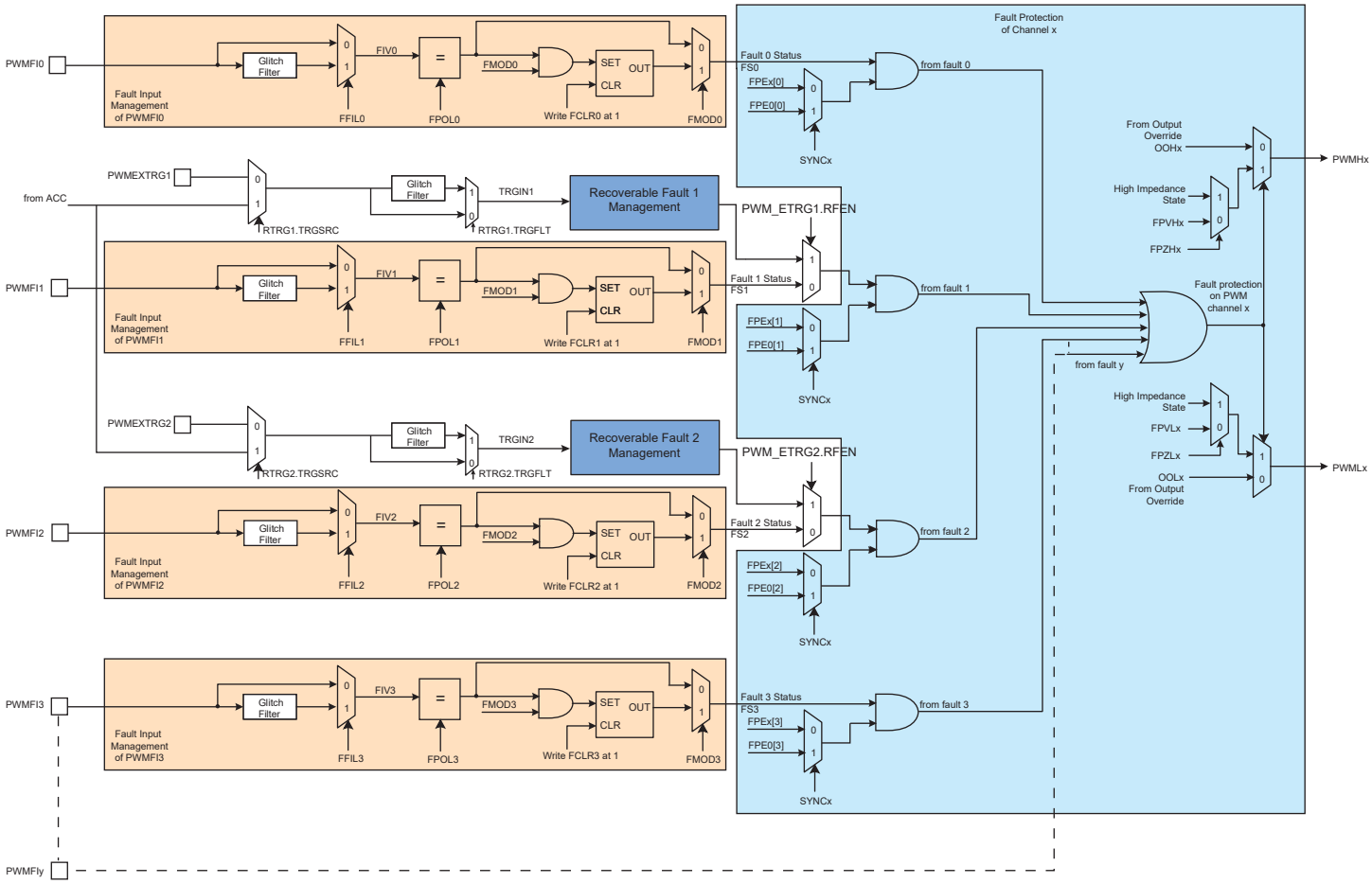
The value of the current output selection can be read in PWM_OS.

While overriding PWM outputs, the channel counters continue to run, only the PWM outputs are forced to user defined values.

53.6.2.7 Fault Protection

6 inputs provide fault protection which can force any of the PWM output pairs to a programmable value. This mechanism has priority over output overriding.

Figure 53-16. Fault Protection



The polarity level of the fault inputs is configured by the FPOL field in the PWM Fault Mode Register (PWM_FMR). For fault inputs coming from internal peripherals such as ADC or Timer Counter, the polarity level must be FPOL = 1. For fault inputs coming from external GPIO pins the polarity level depends on the user's implementation.

The configuration of the Fault Activation mode (FMOD field in PWMC_FMR) depends on the peripheral generating the fault. If the corresponding peripheral does not have “Fault Clear” management, then the FMOD configuration to use must be FMOD = 1, to avoid spurious fault detection. Refer to the corresponding peripheral documentation for details on handling fault generation.

Fault inputs may or may not be glitch-filtered depending on the FFIL field in PWM_FMR. When the filter is activated, glitches on fault inputs with a width inferior to the PWM peripheral clock period are rejected.

A fault becomes active as soon as its corresponding fault input has a transition to the programmed polarity level. If the corresponding bit FMOD is set to ‘0’ in PWM_FMR, the fault remains active as long as the fault input is at this polarity level. If the corresponding FMOD field is set to ‘1’, the fault remains active until the fault input is no longer at this polarity level and until it is cleared by writing the corresponding bit FCLR in the PWM Fault Clear Register (PWM_FCR). In the PWM Fault Status Register (PWM_FSR), the field FIV indicates the current level of the fault inputs and the field FIS indicates whether a fault is currently active.

Each fault can be taken into account or not by the fault protection mechanism in each channel. To be taken into account in the channel x, the fault y must be enabled by the bit FPEx[y] in the PWM Fault Protection Enable registers (PWM_FPE1). However, synchronous channels (see Section 53.6.2.9 “Synchronous Channels”) do not use their own fault enable bits, but those of the channel 0 (bits FPE0[y]).

The fault protection on a channel is triggered when this channel is enabled and when any one of the faults that are enabled for this channel is active. It can be triggered even if the PWM peripheral clock is not running but only by a fault input that is not glitch-filtered.

When the fault protection is triggered on a channel, the fault protection mechanism resets the counter of this channel and forces the channel outputs to the values defined by the fields FPVHx and FPVLx in the PWM Fault Protection Value Register 1 (PWM_FPV) and fields FPZHx/FPZLx in the PWM Fault Protection Value Register 2, as shown in Table 53-5. The output forcing is made asynchronously to the channel counter.

Table 53-5. Forcing Values of PWM Outputs by Fault Protection

FPZH/Lx	FPVH/Lx	Forcing Value of PWMH/Lx
0	0	0
0	1	1
1	–	High impedance state (Hi-Z)

CAUTION:

- To prevent any unexpected activation of the status flag FSy in PWM_FSR, the FMODy bit can be set to ‘1’ only if the FPOLy bit has been previously configured to its final value.
- To prevent any unexpected activation of the Fault Protection on the channel x, the bit FPEx[y] can be set to ‘1’ only if the FPOLy bit has been previously configured to its final value.

If a comparison unit is enabled (see Section 53.6.3 “PWM Comparison Units”) and if a fault is triggered in the channel 0, then the comparison cannot match.

As soon as the fault protection is triggered on a channel, an interrupt (different from the interrupt generated at the end of the PWM period) can be generated but only if it is enabled and not masked. The interrupt is reset by reading the interrupt status register, even if the fault which has caused the trigger of the fault protection is kept active.

Recoverable Fault

The PWM provides a Recoverable Fault mode on fault 1 and 2 (see Figure 53-16).

The recoverable fault signal is an internal signal generated as soon as an external trigger event occurs (see Section 53.6.5 “PWM External Trigger Mode”).

When the fault 1 or 2 is defined as a recoverable fault, the corresponding fault input pin is ignored and bits FFIL1/2, FMOD1/2 and FFIL1/2 are not taken into account.

The fault 1 is managed as a recoverable fault by the PWMEXTRG1 input trigger when PWM_ETRG1.RFEN = 1, PWM_ENA.CHID1 = 1, and PWM_ETRG1.TRGMODE ≠ 0.

The fault 2 is managed as a recoverable fault by the PWMEXTRG2 input trigger when PWM_ETRG2.RFEN = 1, PWM_ENA.CHID2 = 1, and PWM_ETRG2.TRGMODE ≠ 0.

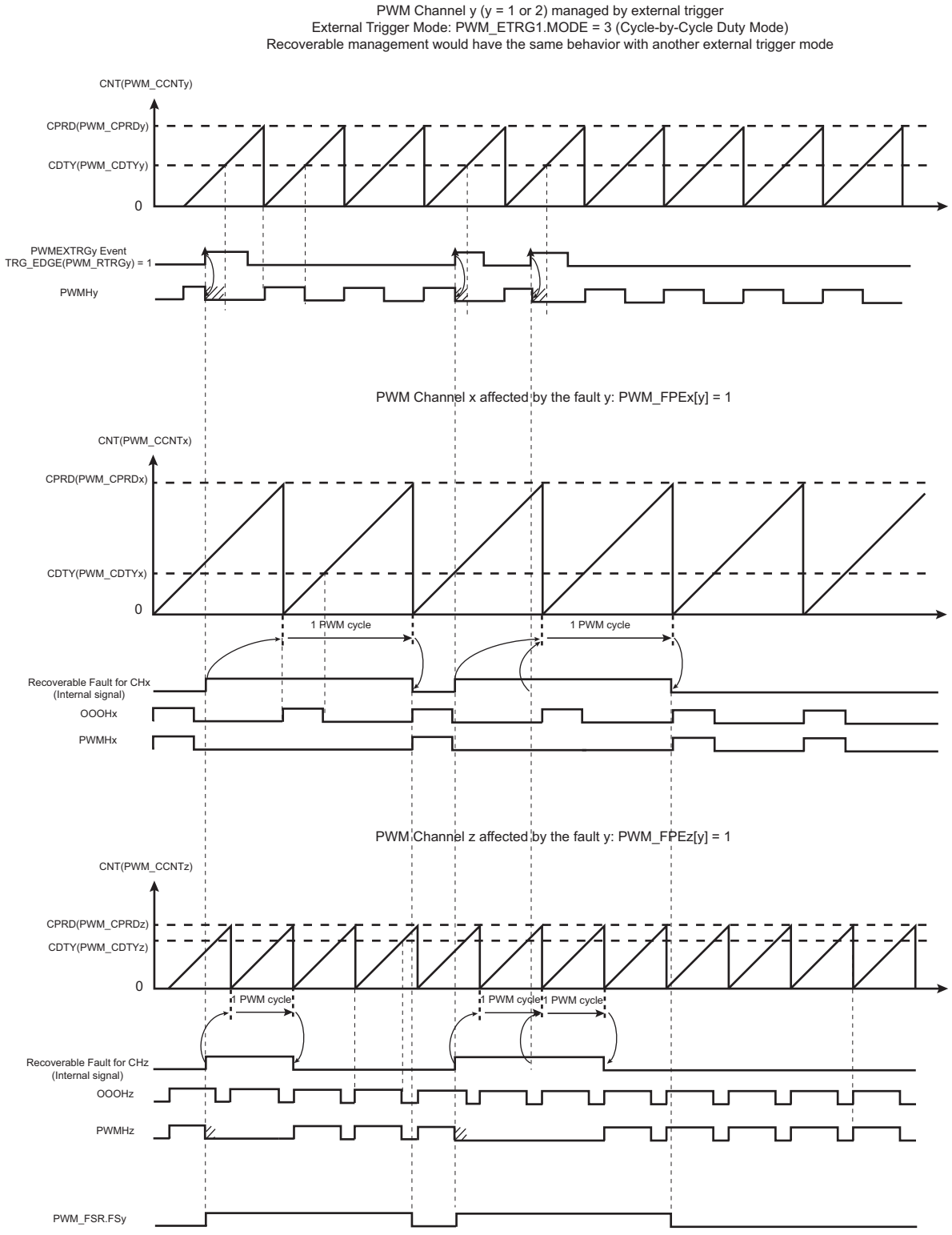
Recoverable fault 1 and 2 can be taken into account by all channels by enabling the bit FPEx[1/2] in the PWM Fault Protection Enable registers (PWM_FPEx). However the synchronous channels (see Section 53.6.2.9 “Synchronous Channels”) do not use their own fault enable bits, but those of the channel 0 (bits FPE0[1/2]).

When a recoverable fault is triggered (according to the PWM_ETRGx.TRGMODE setting), the PWM counter of the affected channels is not cleared (unlike in the classic fault protection mechanism) but the channel outputs are forced to the values defined by the fields FPVHx and FPVLx in the PWM Fault Protection Value Register 1 (PWM_FPV), as per Table 53-5. The output forcing is made asynchronously to the channel counter and lasts from

the recoverable fault occurrence to the end of the next PWM cycle (if the recoverable fault is no longer present) (see Figure 53-17).

The recoverable fault does not trigger an interrupt. The Fault Status FSy (with y = 1 or 2) is not reported in the PWM Fault Status Register when the fault y is a recoverable fault.

Figure 53-17. Recoverable Fault Management



53.6.2.8 Spread Spectrum Counter

The PWM macrocell includes a spread spectrum counter allowing the generation of a constantly varying duty cycle on the output PWM waveform (only for the channel 0). This feature may be useful to minimize electromagnetic interference or to reduce the acoustic noise of a PWM driven motor.

This is achieved by varying the effective period in a range defined by a spread spectrum value which is programmed by the field SPRD in the PWM Spread Spectrum Register (PWM_SSPR). The effective period of the output waveform is the value of the spread spectrum counter added to the programmed waveform period CPRD in the PWM Channel Period Register (PWM_CPRD0).

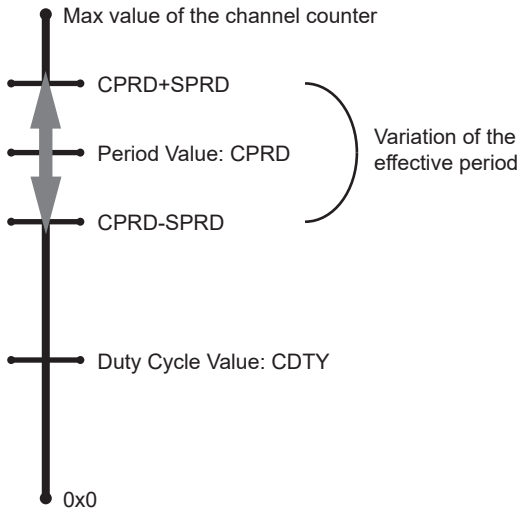
It will cause the effective period to vary from CPRD-SPRD to CPRD+SPRD. This leads to a constantly varying duty cycle on the PWM output waveform because the duty cycle value programmed is unchanged.

The value of the spread spectrum counter can change in two ways depending on the bit SPRDM in PWM_SSPR.

If SPRDM = 0, the Triangular mode is selected. The spread spectrum counter starts to count from -SPRD when the channel 0 is enabled or after reset and counts upwards at each period of the channel counter. When it reaches SPRD, it restarts to count from -SPRD again.

If SPRDM = 1, the Random mode is selected. A new random value is assigned to the spread spectrum counter at each period of the channel counter. This random value is between -SPRD and +SPRD and is uniformly distributed.

Figure 53-18. Spread Spectrum Counter



53.6.2.9 Synchronous Channels

Some channels can be linked together as synchronous channels. They have the same source clock, the same period, the same alignment and are started together. In this way, their counters are synchronized together.

The synchronous channels are defined by the SYNCx bits in the PWM Sync Channels Mode Register (PWM_SCM). Only one group of synchronous channels is allowed.

When a channel is defined as a synchronous channel, the channel 0 is also automatically defined as a synchronous channel. This is because the channel 0 counter configuration is used by all the synchronous channels.

If a channel x is defined as a synchronous channel, the fields/bits for the channel 0 are used instead of those of channel x:

- CPRE in PWM_CMR0 instead of CPRE in PWM_CMRx (same source clock)
- CPRD in PWM_CPRD0 instead of CPRD in PWM_CPRDx (same period)
- CALG in PWM_CMR0 instead of CALG in PWM_CMRx (same alignment)

Modifying the fields CPRE, CPRD and CALG of for channels with index greater than 0 has no effect on output waveforms.

Because counters of synchronous channels must start at the same time, they are all enabled together by enabling the channel 0 (by the CHID0 bit in PWM_ENA register). In the same way, they are all disabled together by disabling channel 0 (by the CHID0 bit in PWM_DIS register). However, a synchronous channel x different from channel 0 can be enabled or disabled independently from others (by the CHIDx bit in PWM_ENA and PWM_DIS registers).

Defining a channel as a synchronous channel while it is an asynchronous channel (by writing the bit SYNCx to '1' while it was at '0') is allowed only if the channel is disabled at this time (CHIDx = 0 in PWM_SR). In the same way, defining a channel as an asynchronous channel while it is a synchronous channel (by writing the SYNCx bit to '0' while it was '1') is allowed only if the channel is disabled at this time.

The UPDM field (Update Mode) in the PWM_SCM register selects one of the three methods to update the registers of the synchronous channels:

- Method 1 (UPDM = 0): The period value, the duty-cycle values and the dead-time values must be written by the processor in their respective update registers (respectively PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPDx).The update is triggered at the next PWM period as soon as the bit UPDULOCK in the PWM Sync Channels Update Control Register (PWM_SCUC) is set to '1' (see "Method 1: Manual write of duty-cycle values and manual trigger of the update").
- Method 2 (UPDM = 1): The period value, the duty-cycle values, the dead-time values and the update period value must be written by the processor in their respective update registers (respectively PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPD). The update of the period value and of the dead-time values is triggered at the next PWM period as soon as the bit UPDULOCK in the PWM_SCUC register is set to '1'. The update of the duty-cycle values and the update period value is triggered automatically after an update period defined by the field UPR in the PWM Sync Channels Update Period Register (PWM_SCUP) (see "Method 2: Manual write of duty-cycle values and automatic trigger of the update").
- Method 3 (UPDM = 2): Same as Method 2 apart from the fact that the duty-cycle values of ALL synchronous channels are written by the DMA Controller (see "Method 3: Automatic write of duty-cycle values and automatic trigger of the update"). The user can choose to synchronize the DMA Controller transfer request with a comparison match (see Section 53.6.3 "PWM Comparison Units"), by the fields PTRM and PTRCS in the PWM_SCM register. The DMA destination address must be configured to access only the PWM DMA Register (PWM_DMAR). The DMA buffer data structure must consist of sequentially repeated duty cycles. The number of duty cycles in each sequence corresponds to the number of synchronized channels. Duty cycles in each sequence must be ordered from the lowest to the highest channel index. The size of the duty cycle is 16 bits.

Table 53-6. Summary of the Update of Registers of Synchronous Channels

Register	UPDM = 0	UPDM = 1	UPDM = 2
Period Value (PWM_CPRDUPDx)	Write by the processor		
	Update is triggered at the next PWM period as soon as the bit UPDULOCK is set to '1'		
Dead-Time Values (PWM_DTUPDx)	Write by the processor		
	Update is triggered at the next PWM period as soon as the bit UPDULOCK is set to '1'		
Duty-Cycle Values (PWM_CDTYUPDx)	Write by the processor	Write by the processor	Write by the DMA Controller
	Update is triggered at the next PWM period as soon as the bit UPDULOCK is set to '1'	Update is triggered at the next PWM period as soon as the update period counter has reached the value UPR	

Table 53-6. Summary of the Update of Registers of Synchronous Channels (Continued)

Register	UPDM = 0	UPDM = 1	UPDM = 2
Update Period Value (PWM_SCUPUPD)	Not applicable	Write by the processor	
	Not applicable	Update is triggered at the next PWM period as soon as the update period counter has reached the value UPR	

Method 1: Manual write of duty-cycle values and manual trigger of the update

In this mode, the update of the period value, the duty-cycle values and the dead-time values must be done by writing in their respective update registers with the processor (respectively PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPDx).

To trigger the update, the user must use the bit UPDULOCK in the PWM_SCUC register which allows to update synchronously (at the same PWM period) the synchronous channels:

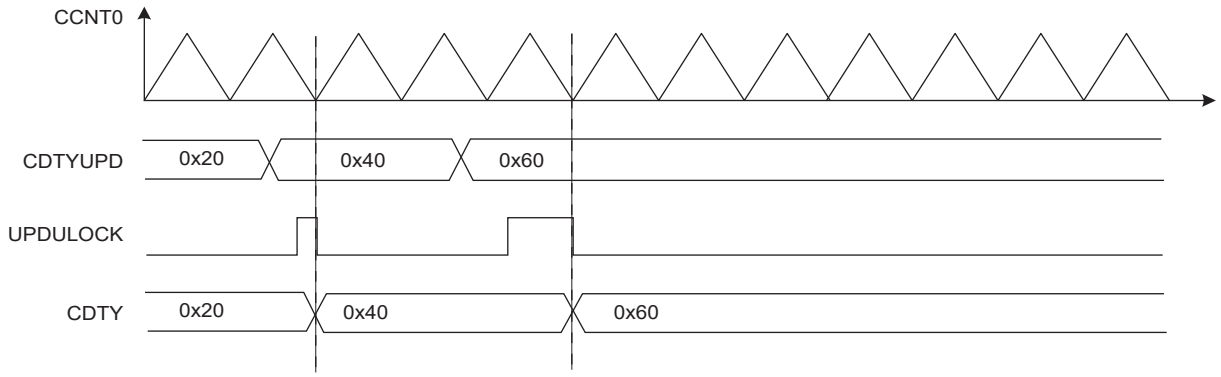
- If the bit UPDULOCK is set to ‘1’, the update is done at the next PWM period of the synchronous channels.
- If the UPDULOCK bit is not set to ‘1’, the update is locked and cannot be performed.

After writing the UPDULOCK bit to ‘1’, it is held at this value until the update occurs, then it is read 0.

Sequence for Method 1:

1. Select the manual write of duty-cycle values and the manual update by setting the UPDM field to ‘0’ in the PWM_SCM register.
2. Define the synchronous channels by the SYNCx bits in the PWM_SCM register.
3. Enable the synchronous channels by writing CHID0 in the PWM_ENA register.
4. If an update of the period value and/or the duty-cycle values and/or the dead-time values is required, write registers that need to be updated (PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPDx).
5. Set UPDULOCK to ‘1’ in PWM_SCUC.
6. The update of the registers will occur at the beginning of the next PWM period. When the UPDULOCK bit is reset, go to Step 4. for new values.

Figure 53-19. Method 1 (UPDM = 0)



Method 2: Manual write of duty-cycle values and automatic trigger of the update

In this mode, the update of the period value, the duty-cycle values, the dead-time values and the update period value must be done by writing in their respective update registers with the processor (respectively PWM_CPRDUPDx, PWM_CDTYUPDx, PWM_DTUPDx and PWM_SCUPUPD).

To trigger the update of the period value and the dead-time values, the user must use the bit UPDULOCK in the PWM_SCUC register, which updates synchronously (at the same PWM period) the synchronous channels:

- If the bit UPDULOCK is set to ‘1’, the update is done at the next PWM period of the synchronous channels.
- If the UPDULOCK bit is not set to ‘1’, the update is locked and cannot be performed.

After writing the UPDULOCK bit to ‘1’, it is held at this value until the update occurs, then it is read 0.

The update of the duty-cycle values and the update period is triggered automatically after an update period.

To configure the automatic update, the user must define a value for the update period by the UPR field in the PWM_SCUP register. The PWM controller waits UPR+1 period of synchronous channels before updating automatically the duty values and the update period value.

The status of the duty-cycle value write is reported in the PWM Interrupt Status Register 2 (PWM_ISR2) by the following flags:

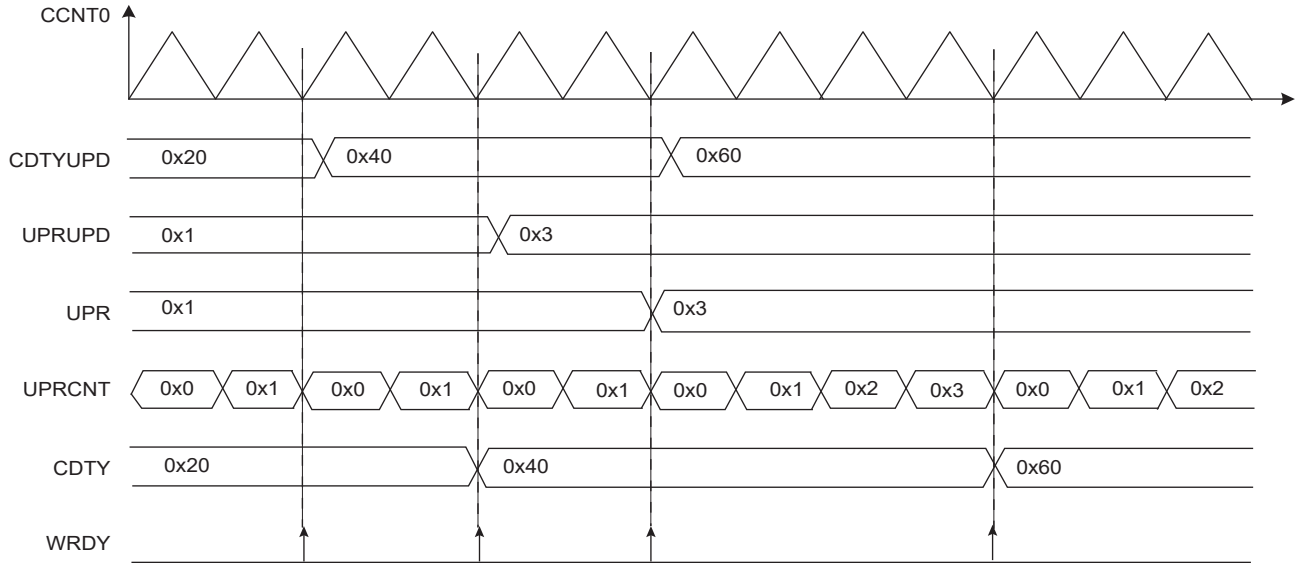
- WRDY: this flag is set to ‘1’ when the PWM Controller is ready to receive new duty-cycle values and a new update period value. It is reset to ‘0’ when the PWM_ISR2 register is read.

Depending on the interrupt mask in the PWM Interrupt Mask Register 2 (PWM_IMR2), an interrupt can be generated by these flags.

Sequence for Method 2:

1. Select the manual write of duty-cycle values and the automatic update by setting the field UPDM to ‘1’ in the PWM_SCM register
2. Define the synchronous channels by the bits SYNCx in the PWM_SCM register.
3. Define the update period by the field UPR in the PWM_SCUP register.
4. Enable the synchronous channels by writing CHID0 in the PWM_ENA register.
5. If an update of the period value and/or of the dead-time values is required, write registers that need to be updated (PWM_CPRDUPDx, PWM_DTUPDx), else go to Step 8.
6. Set UPDULOCK to ‘1’ in PWM_SCUC.
7. The update of these registers will occur at the beginning of the next PWM period. At this moment the bit UPDULOCK is reset, go to Step 5. for new values.
8. If an update of the duty-cycle values and/or the update period is required, check first that write of new update values is possible by polling the flag WRDY (or by waiting for the corresponding interrupt) in PWM_ISR2.
9. Write registers that need to be updated (PWM_CDTYUPDx, PWM_SCUPUPD).
10. The update of these registers will occur at the next PWM period of the synchronous channels when the Update Period is elapsed. Go to Step 8. for new values.

Figure 53-20. Method 2 (UPDM = 1)



Method 3: Automatic write of duty-cycle values and automatic trigger of the update

In this mode, the update of the duty cycle values is made automatically by the DMA Controller. The update of the period value, the dead-time values and the update period value must be done by writing in their respective update registers with the processor (respectively PWM_CPRDUPDx, PWM_DTUPDx and PWM_SCUPUPD).

To trigger the update of the period value and the dead-time values, the user must use the bit UPDULOCK which allows to update synchronously (at the same PWM period) the synchronous channels:

- If the bit UPDULOCK is set to ‘1’, the update is done at the next PWM period of the synchronous channels.
- If the UPDULOCK bit is not set to ‘1’, the update is locked and cannot be performed.

After writing the UPDULOCK bit to ‘1’, it is held at this value until the update occurs, then it is read 0.

The update of the duty-cycle values and the update period value is triggered automatically after an update period.

To configure the automatic update, the user must define a value for the Update Period by the field UPR in the PWM_SCUP register. The PWM controller waits UPR+1 periods of synchronous channels before updating automatically the duty values and the update period value.

Using the DMA Controller removes processor overhead by reducing its intervention during the transfer. This significantly reduces the number of clock cycles required for a data transfer, which improves microcontroller performance.

The DMA Controller must write the duty-cycle values in the synchronous channels index order. For example if the channels 0, 1 and 3 are synchronous channels, the DMA Controller must write the duty-cycle of the channel 0 first, then the duty-cycle of the channel 1, and finally the duty-cycle of the channel 3.

The status of the DMA Controller transfer is reported in PWM_ISR2 by the following flags:

- WRDY: this flag is set to ‘1’ when the PWM Controller is ready to receive new duty-cycle values and a new update period value. It is reset to ‘0’ when PWM_ISR2 is read. The user can choose to synchronize the WRDY flag and the DMA Controller transfer request with a comparison match (see Section 53.6.3 “PWM Comparison Units”), by the fields PTRM and PTRCS in the PWM_SCM register.
- UNRE: this flag is set to ‘1’ when the update period defined by the UPR field has elapsed while the whole data has not been written by the DMA Controller. It is reset to ‘0’ when PWM_ISR2 is read.

Depending on the interrupt mask in PWM_IMR2, an interrupt can be generated by these flags.

Sequence for Method 3:

1. Select the automatic write of duty-cycle values and automatic update by setting the field UPDM to 2 in the PWM_SCM register.
2. Define the synchronous channels by the bits SYNCx in the PWM_SCM register.
3. Define the update period by the field UPR in the PWM_SCUP register.
4. Define when the WRDY flag and the corresponding DMA Controller transfer request must be set in the update period by the PTRM bit and the PTRCS field in the PWM_SCM register (at the end of the update period or when a comparison matches).
5. Define the DMA Controller transfer settings for the duty-cycle values and enable it in the DMA Controller registers
6. Enable the synchronous channels by writing CHID0 in the PWM_ENA register.
7. If an update of the period value and/or of the dead-time values is required, write registers that need to be updated (PWM_CPRDUPDx, PWM_DTUPDx), else go to Step 10.
8. Set UPDULOCK to ‘1’ in PWM_SCUC.
9. The update of these registers will occur at the beginning of the next PWM period. At this moment the bit UPDULOCK is reset, go to Step 7. for new values.
10. If an update of the update period value is required, check first that write of a new update value is possible by polling the flag WRDY (or by waiting for the corresponding interrupt) in PWM_ISR2, else go to Step 13.
11. Write the register that needs to be updated (PWM_SCUPUPD).

12. The update of this register will occur at the next PWM period of the synchronous channels when the Update Period is elapsed. Go to Step 10. for new values.
13. Wait for the DMA status flag indicating that the buffer transfer is complete. If the transfer has ended, define a new DMA transfer for new duty-cycle values. Go to Step 5.

Figure 53-21. Method 3 (UPDM = 2 and PTRM = 0)

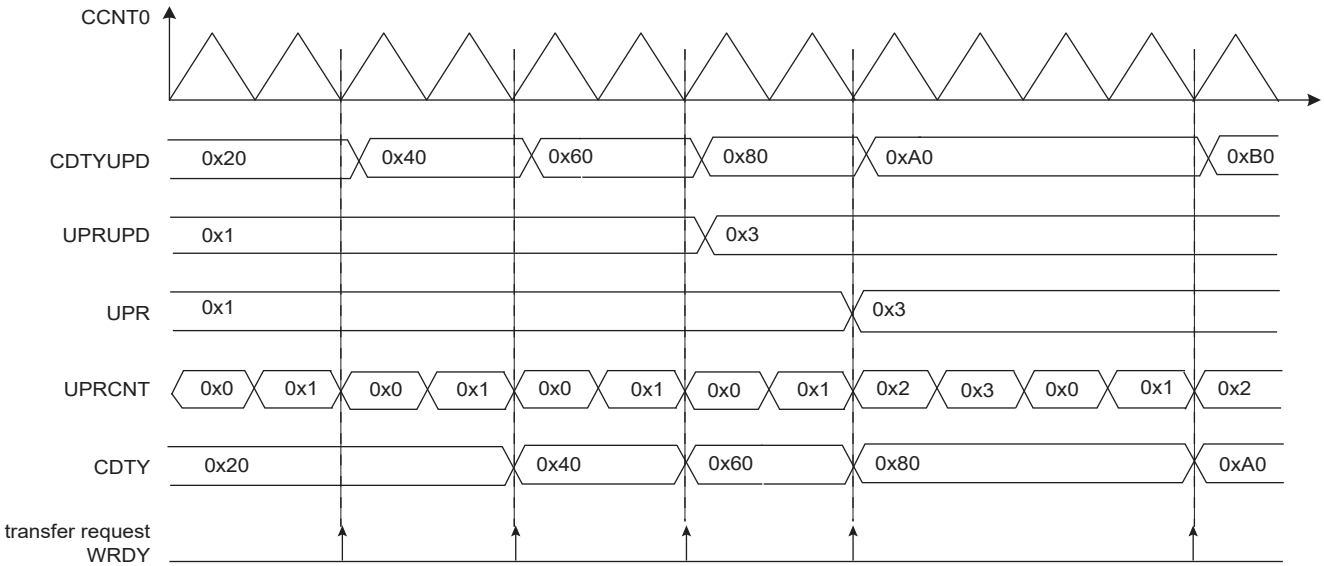
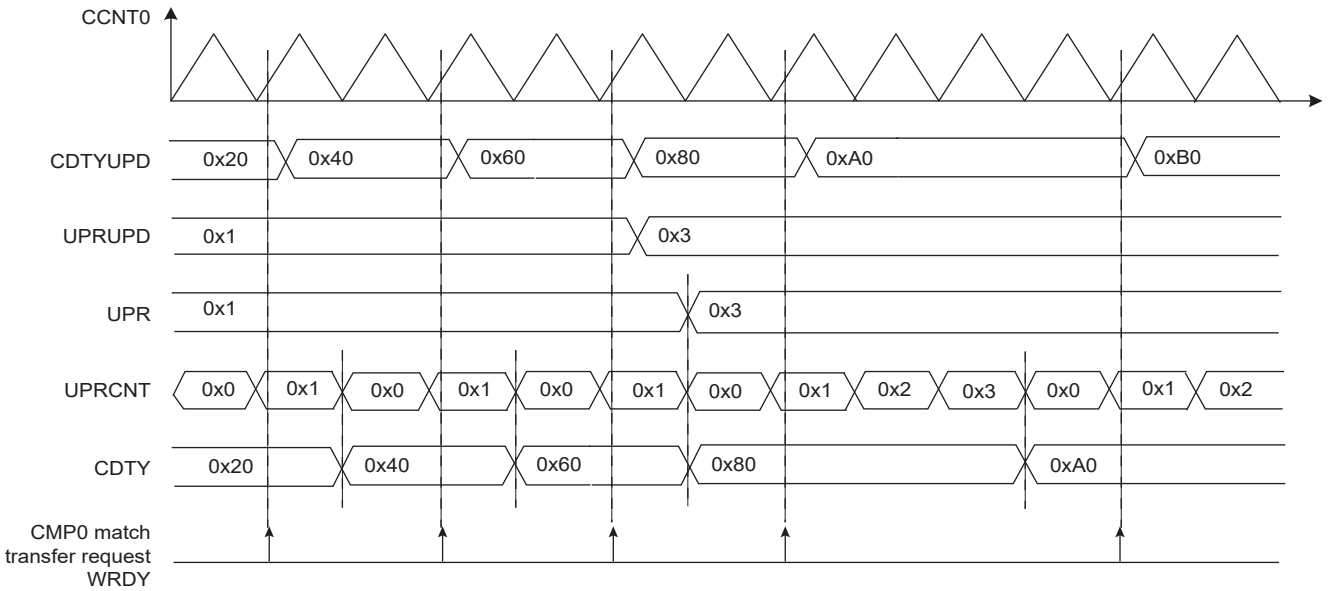


Figure 53-22. Method 3 (UPDM = 2 and PTRM = 1 and PTRCS = 0)



53.6.2.10 Update Time for Double-Buffering Registers

All channels integrate a double-buffering system in order to prevent an unexpected output waveform while modifying the period, the spread spectrum value, the polarity, the duty-cycle, the dead-times, the output override, and the synchronous channels update period.

This double-buffering system comprises the following update registers:

- PWM Sync Channels Update Period Update Register
- PWM Output Selection Set Update Register
- PWM Output Selection Clear Update Register
- PWM Spread Spectrum Update Register
- PWM Channel Duty Cycle Update Register
- PWM Channel Period Update Register
- PWM Channel Dead Time Update Register
- PWM Channel Mode Update Register

When one of these update registers is written to, the write is stored, but the values are updated only at the next PWM period border. In Left-aligned mode (CALG = 0), the update occurs when the channel counter reaches the period value CPRD. In Center-aligned mode, the update occurs when the channel counter value is decremented and reaches the 0 value.

In Center-aligned mode, it is possible to trigger the update of the polarity and the duty-cycle at the next half period border. This mode concerns the following update registers:

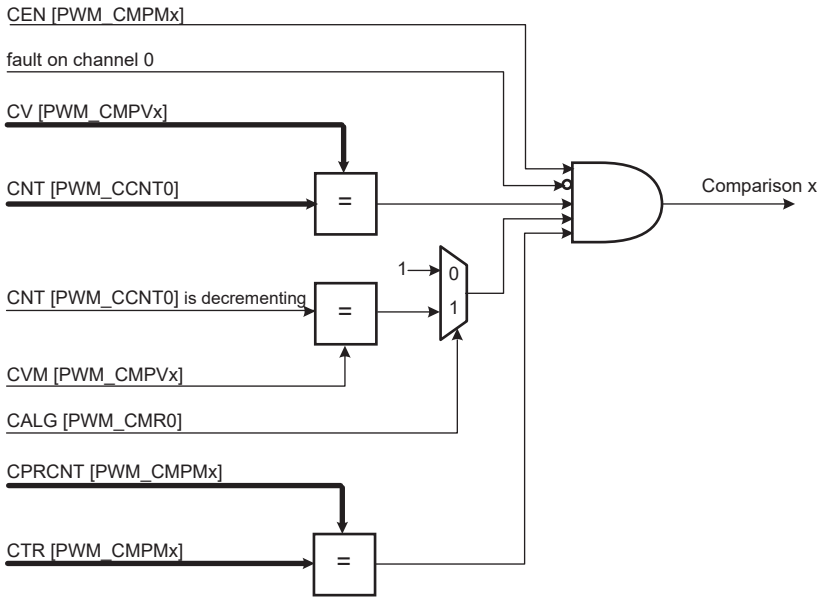
- PWM Channel Duty Cycle Update Register
- PWM Channel Mode Update Register

The update occurs at the first half period following the write of the update register (either when the channel counter value is incrementing and reaches the period value CPRD, or when the channel counter value is decrementing and reaches the 0 value). To activate this mode, the user must write a one to the bit UPDS in the PWM Channel Mode Register.

53.6.3 PWM Comparison Units

The PWM provides 1 independent comparison units able to compare a programmed value with the current value of the channel 0 counter (which is the channel counter of all synchronous channels, Section 53.6.2.9 “Synchronous Channels”). These comparisons are intended to generate pulses on the event lines (used to synchronize ADC, see Section 53.6.4 “PWM Event Lines”), to generate software interrupts and to trigger DMA Controller transfer requests for the synchronous channels (see “Method 3: Automatic write of duty-cycle values and automatic trigger of the update”).

Figure 53-23. Comparison Unit Block Diagram



The comparison x matches when it is enabled by the bit CEN in the PWM Comparison x Mode Register (PWM_CMPMx for the comparison x) and when the counter of the channel 0 reaches the comparison value defined by the field CV in PWM Comparison x Value Register (PWM_CMPVx for the comparison x). If the counter of the channel 0 is center-aligned (CALG = 1 in PWM Channel Mode Register), the bit CVM in PWM_CMPVx defines if the comparison is made when the counter is counting up or counting down (in Left-alignment mode CALG = 0, this bit is useless).

If a fault is active on the channel 0, the comparison is disabled and cannot match (see Section 53.6.2.7 “Fault Protection”).

The user can define the periodicity of the comparison x by the fields CTR and CPR in PWM_CMPMx. The comparison is performed periodically once every CPR+1 periods of the counter of the channel 0, when the value of the comparison period counter CPRCNT in PWM_CMPMx reaches the value defined by CTR. CPR is the maximum value of the comparison period counter CPRCNT. If CPR = CTR = 0, the comparison is performed at each period of the counter of the channel 0.

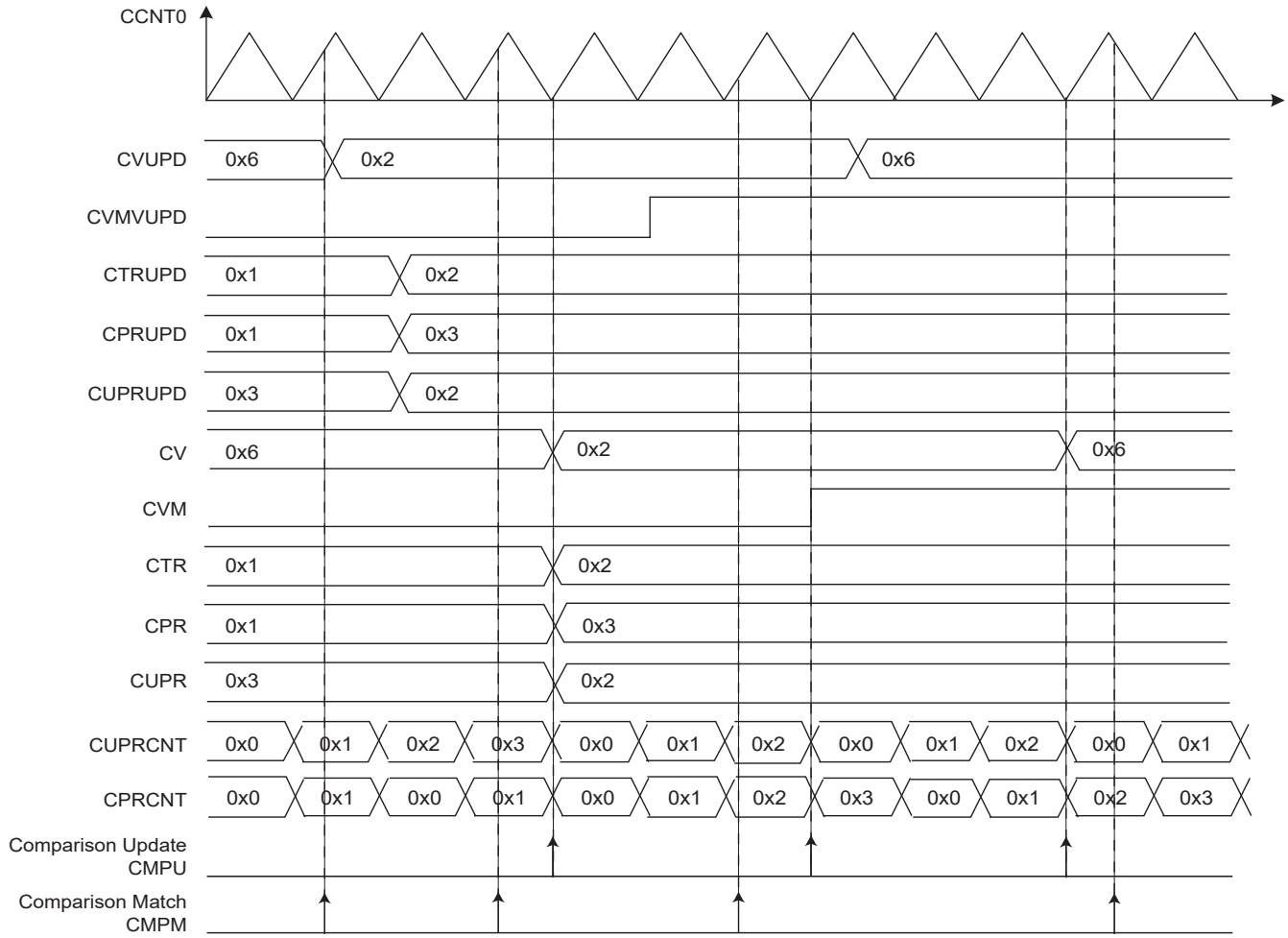
The comparison x configuration can be modified while the channel 0 is enabled by using the PWM Comparison x Mode Update Register (PWM_CMPMUPDx registers for the comparison x). In the same way, the comparison x value can be modified while the channel 0 is enabled by using the PWM Comparison x Value Update Register (PWM_CMPVUPDx registers for the comparison x).

The update of the comparison x configuration and the comparison x value is triggered periodically after the comparison x update period. It is defined by the field CUPR in PWM_CMPMx. The comparison unit has an update period counter independent from the period counter to trigger this update. When the value of the comparison update period counter CUPRCNT (in PWM_CMPMx) reaches the value defined by CUPR, the update is triggered. The comparison x update period CUPR itself can be updated while the channel 0 is enabled by using the PWM_CMPMUPDx register.

CAUTION: The write of PWM_CMPVUPDx must be followed by a write of PWM_CMPMUPDx.

The comparison match and the comparison update can be source of an interrupt, but only if it is enabled and not masked. These interrupts can be enabled by the PWM Interrupt Enable Register 2 and disabled by the PWM Interrupt Disable Register 2. The comparison match interrupt and the comparison update interrupt are reset by reading the PWM Interrupt Status Register 2.

Figure 53-24. Comparison Waveform



53.6.4 PWM Event Lines

The PWM provides 2 independent event lines intended to trigger actions in other peripherals (e.g., for the Analog-to-Digital Converter (ADC)).

A pulse (one cycle of the peripheral clock) is generated on an event line, when at least one of the selected comparisons is matching. The comparisons can be selected or unselected independently by the CSEL bits in the PWM Event Line x Register (PWM_ELMRx for the Event Line x).

An example of event generation is provided in Figure 53-26.

Figure 53-25. Event Line Block Diagram

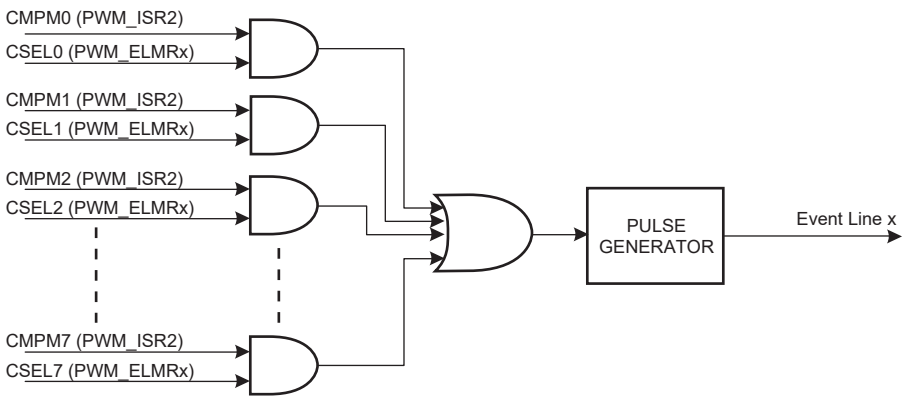
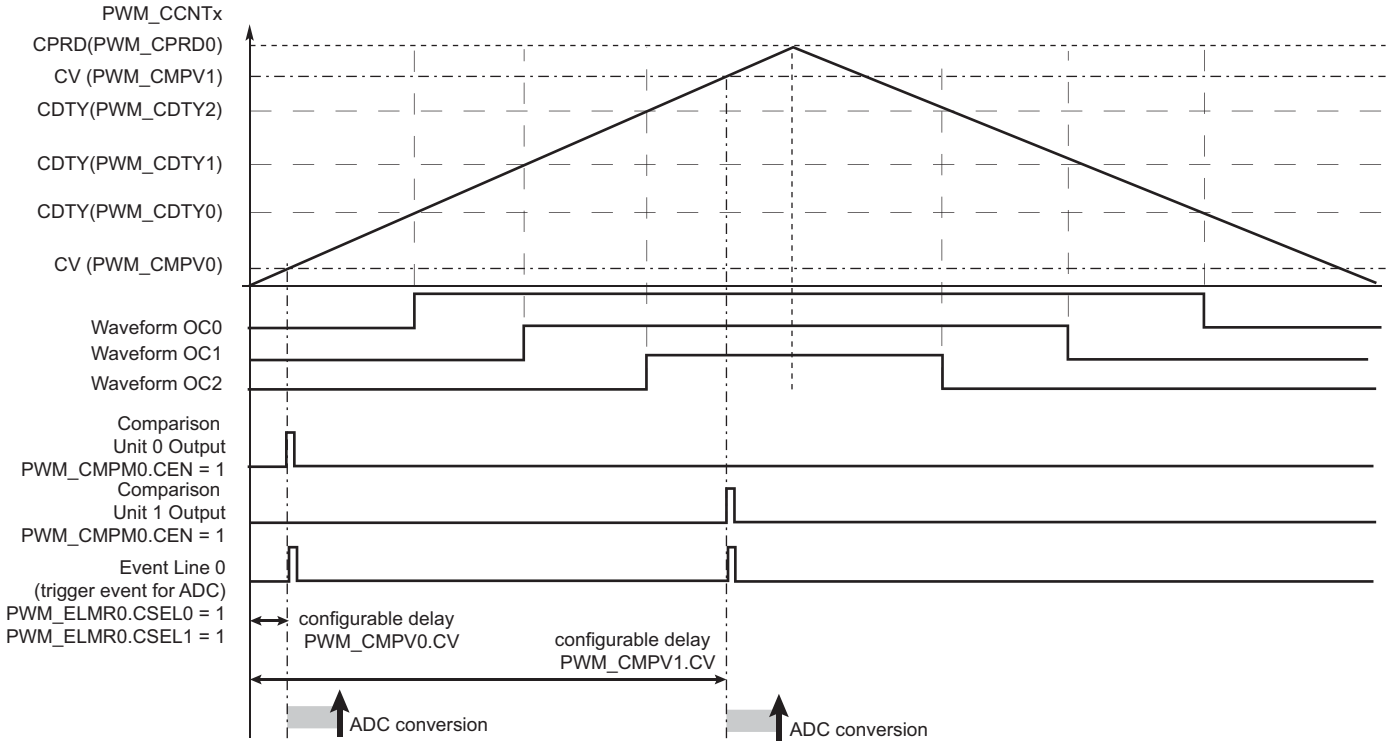


Figure 53-26. Event Line Generation Waveform (Example)



53.6.5 PWM External Trigger Mode

The PWM channels 1 and 2 can be configured to use an external trigger for generating specific PWM signals. The external trigger source can be selected through the bit TRGSRC of the PWM External Trigger Register (see Table 53-7).

Table 53-7. External Event Source Selection

Channel	Trigger Source Selection	Trigger Source
1	PWM_ETRG1.TRGSRC = 0	From PWMEXTRG1 input
	PWM_ETRG1.TRGSRC = 1	From Analog Comparator Controller
2	PWM_ETRG2.TRGSRC = 0	From PWMEXTRG2 input
	PWM_ETRG2.TRGSRC = 1	From Analog Comparator Controller

Each external trigger source can be filtered by writing a one to the TRGFILT bit in the corresponding PWM External Trigger Register (PWM_ETRGx).

Each time an external trigger event is detected, the corresponding PWM channel counter value is stored in the MAXCNT field of the PWM_ETRGx register if it is greater than the previously stored value. Reading the PWM_ETRGx register will clear the MAXCNT value.

Three different modes are available for channels 1 and 2 depending on the value of the TRGMODE field of the PWM_ETRGx register:

- TRGMODE = 1: External PWM Reset Mode (see Section 53.6.5.1 “External PWM Reset Mode”)
- TRGMODE = 2: External PWM Start Mode (see Section 53.6.5.2 “External PWM Start Mode”)
- TRGMODE = 3: Cycle-By-Cycle Duty Mode (see Section 53.6.5.3 “Cycle-By-Cycle Duty Mode”)

This feature is disabled when TRGMODE = 0.

This feature should only be enabled if the corresponding channel is left-aligned (CALG = 0 in PWM Channel Mode Register of channel 1 or 2) and not managed as a synchronous channel (SYNCx = 0 in PWM Sync Channels Mode Register where x = 1 or 2). Programming the channel to be center-aligned or synchronous while TRGMODE is not 0 could lead to unexpected behavior.

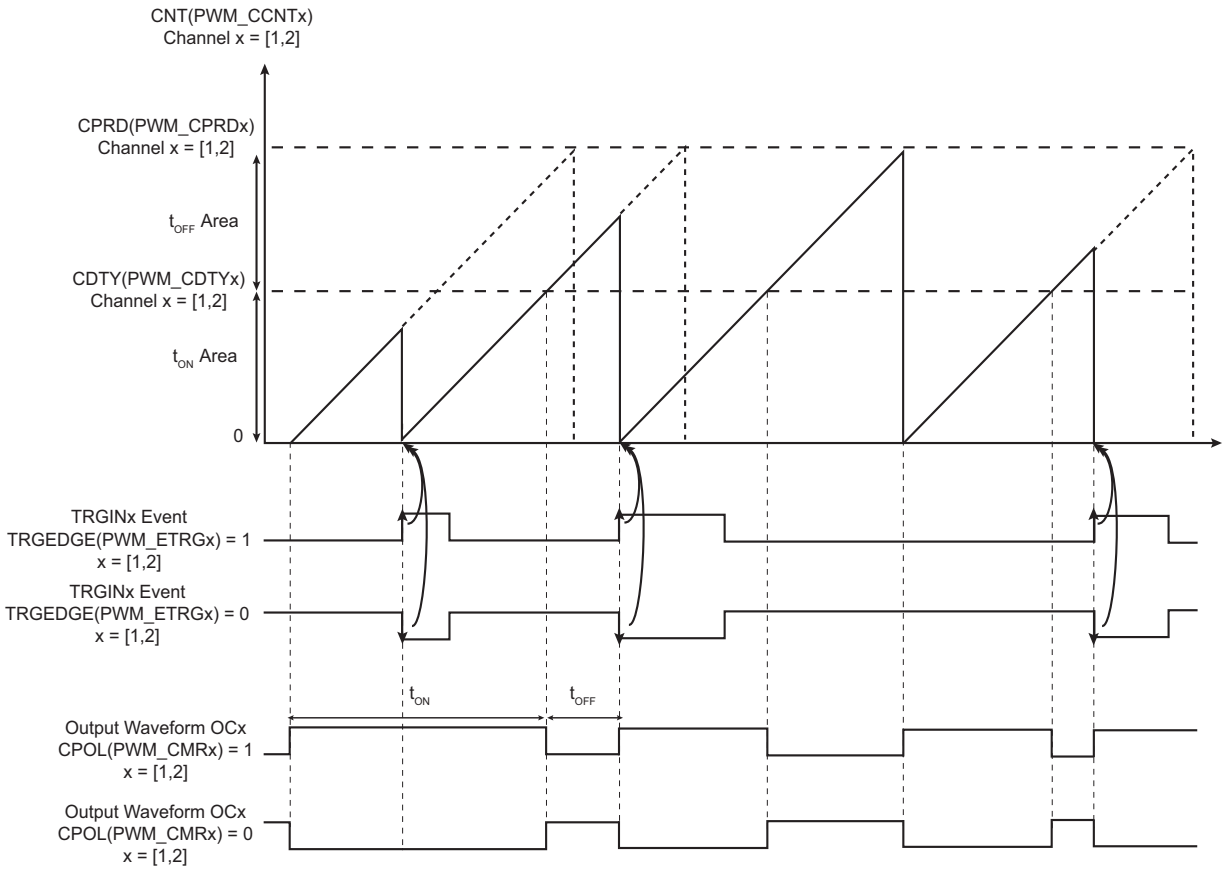
53.6.5.1 External PWM Reset Mode

External PWM Reset mode is selected by programming TRGMODE = 1 in the PWM_ETRGx register.

In this mode, when an edge is detected on the PWMEXTRGx input, the internal PWM counter is cleared and a new PWM cycle is restarted. The edge polarity can be selected by programming the TRGEDGE bit in the PWM_ETRGx register. If no trigger event is detected when the internal channel counter has reached the CPRD value in the PWM Channel Period Register, the internal counter is cleared and a new PWM cycle starts.

Note that this mode does not guarantee a constant t_{ON} or t_{OFF} time.

Figure 53-27. External PWM Reset Mode



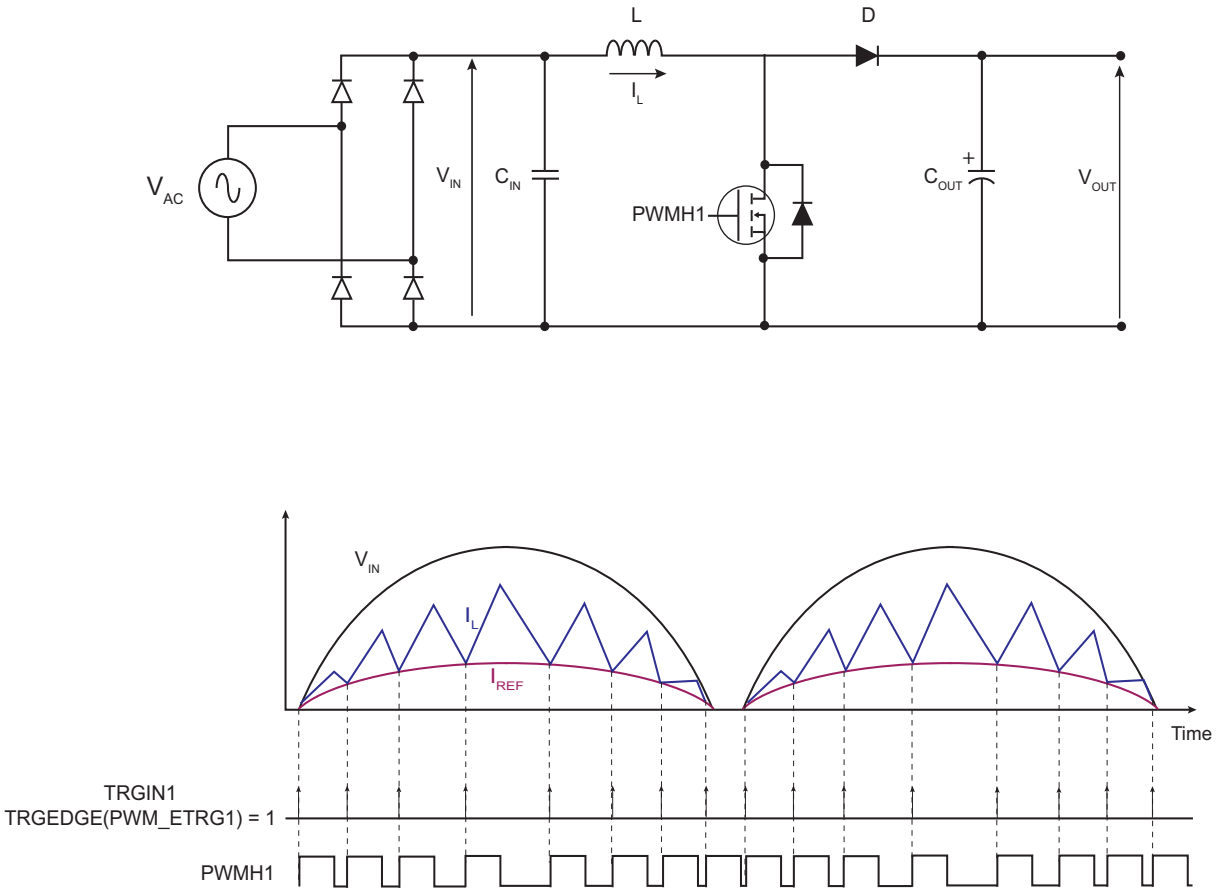
Application Example

The external PWM Reset mode can be used in power factor correction applications.

In the example below, the external trigger input is the PWMEXTRG1 (therefore the PWM channel used for regulation is the channel 1). The PWM channel 1 period (CPRD in the PWM Channel Period Register of the channel 1) must be programmed so that the TRGIN1 event always triggers before the PWM channel 1 period elapses.

In Figure 53-28, an external circuit (not shown) is required to sense the inductor current I_L. The internal PWM counter of the channel 1 is cleared when the inductor current falls below a specific threshold (I_{REF}). This starts a new PWM period and increases the inductor current.

Figure 53-28. External PWM Reset Mode: Power Factor Correction Application



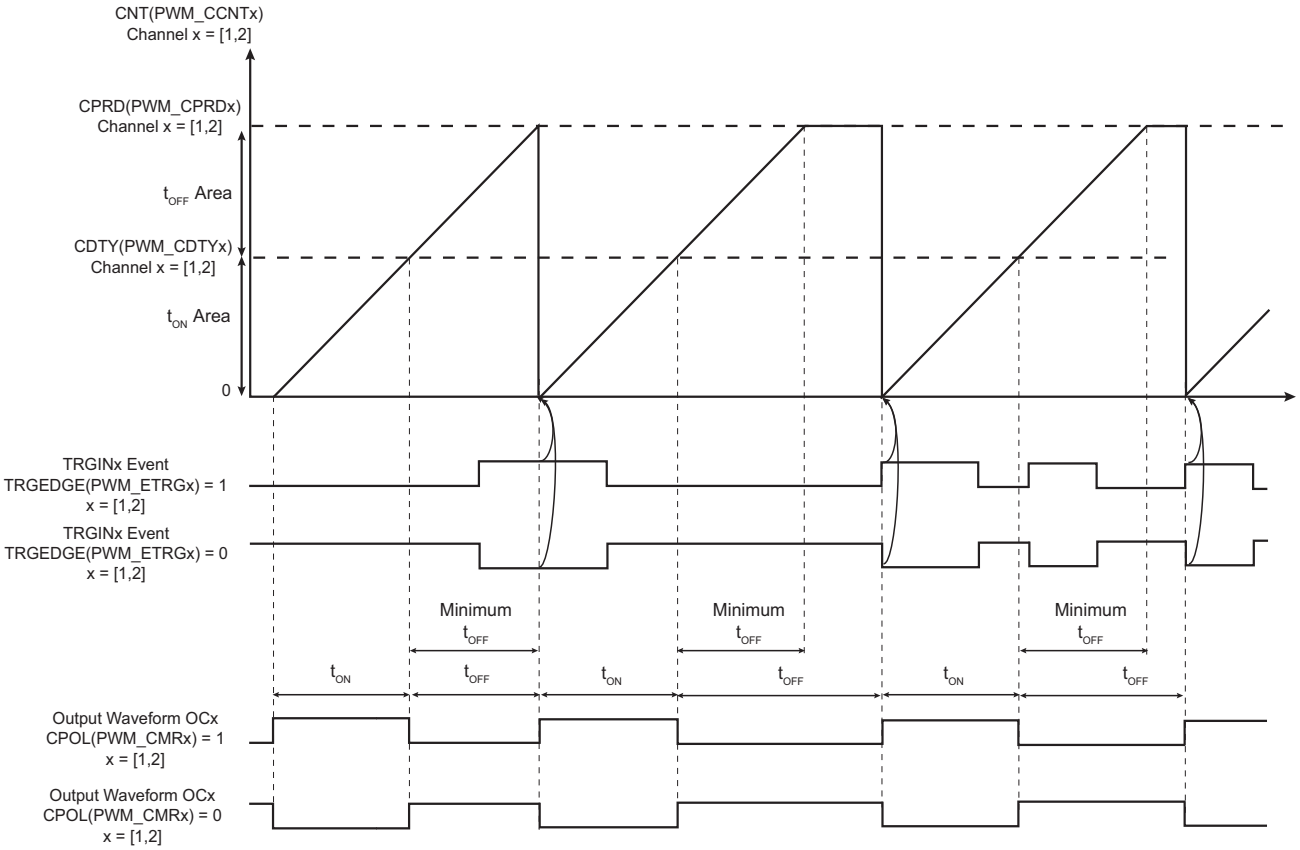
53.6.5.2 External PWM Start Mode

External PWM Start mode is selected by programming TRGMODE = 2 in the PWM_ETRGx register.

In this mode, the internal PWM counter can only be reset once it has reached the CPRD value in the PWM Channel Period Register and when the correct level is detected on the corresponding external trigger input. Both conditions have to be met to start a new PWM period. The active detection level is defined by the bit TRGEDGE of the PWM_ETRGx register.

Note that this mode guarantees a constant t_{ON} time and a minimum t_{OFF} time.

Figure 53-29. External PWM Start Mode



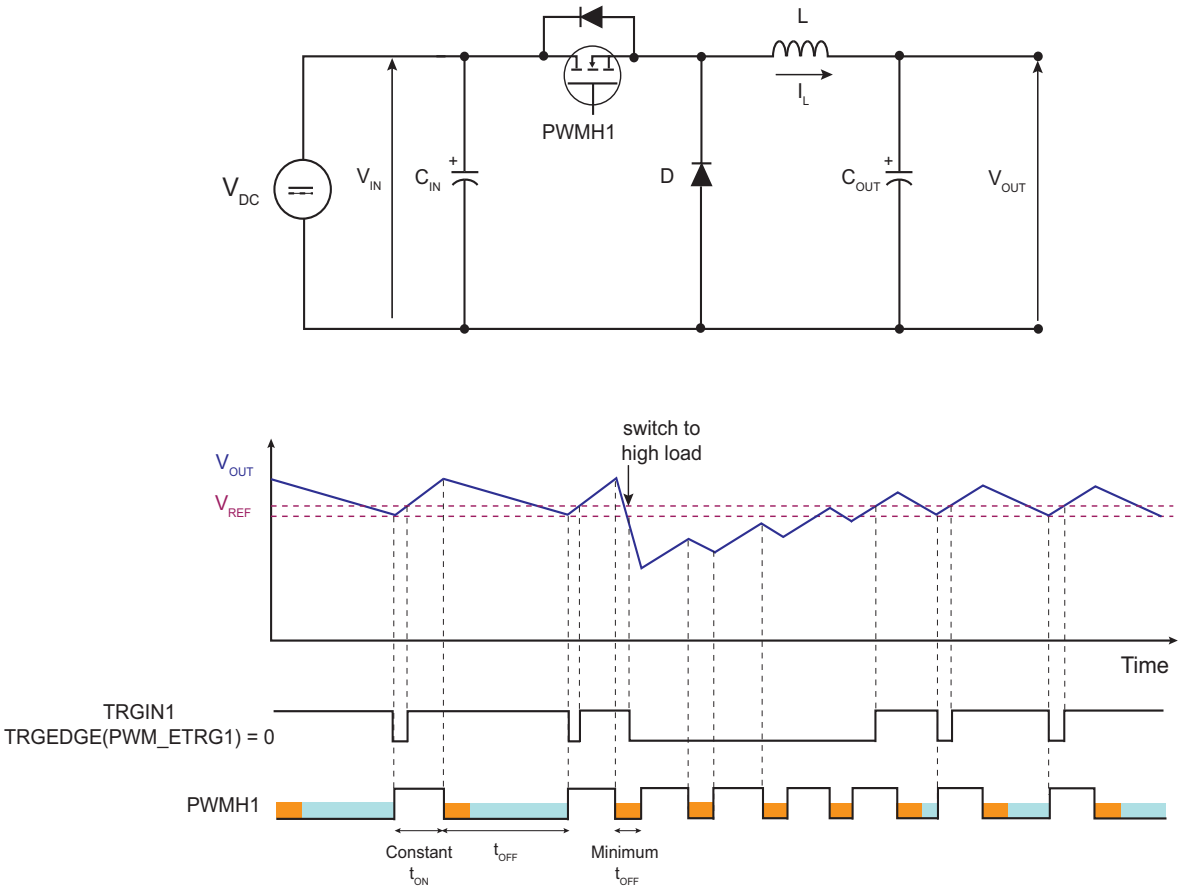
Application Example

The external PWM Start mode generates a modulated frequency PWM signal with a constant active level duration (t_{ON}) and a minimum inactive level duration (minimum t_{OFF}).

The t_{ON} time is defined by the CDTY value in the PWM Channel Duty Cycle Register. The minimum t_{OFF} time is defined by CDTY - CPRD (PWM Channel Period Register). This mode can be useful in Buck DC/DC Converter applications.

When the output voltage V_{OUT} is above a specific threshold (Vref), the PWM inactive level is maintained as long as V_{OUT} remains above this threshold. If V_{OUT} is below this specific threshold, this mode guarantees a minimum t_{OFF} time required for MOSFET driving (see Figure 53-30).

Figure 53-30. External PWM Start Mode: Buck DC/DC Converter



53.6.5.3 Cycle-By-Cycle Duty Mode

Description

Cycle-by-cycle duty mode is selected by programming TRGMODE = 3 in PWM_ETRGx.

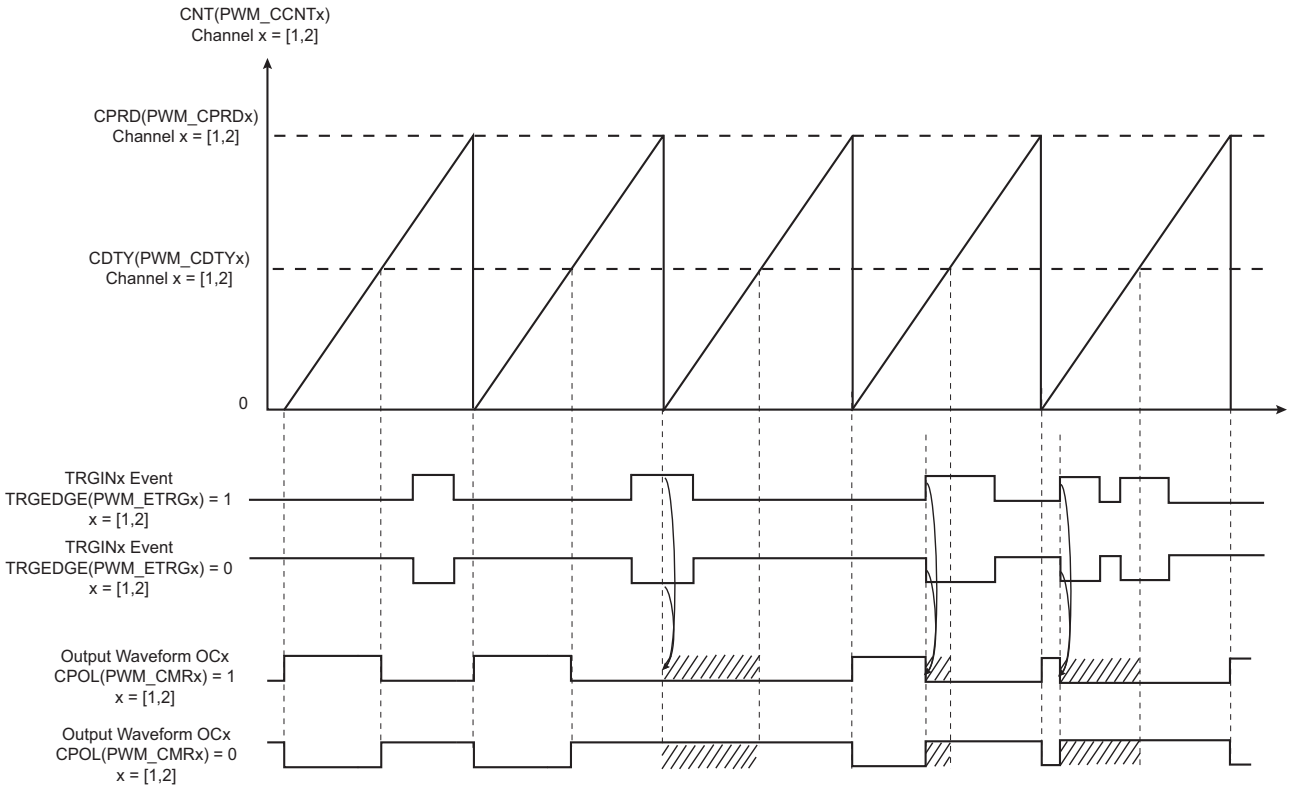
In this mode, the PWM frequency is constant and is defined by the CPRD value in the PWM Channel Period Register.

An external trigger event has no effect on the PWM output if it occurs while the internal PWM counter value is above the CDTY value of the PWM Channel Duty Cycle Register.

If the internal PWM counter value is below the value of CDTY of the PWM Channel Duty Cycle Register, an external trigger event makes the PWM output inactive.

The external trigger event can be detected on rising or falling edge according to the TRGEDGE bit in PWM_ETRGx.

Figure 53-31. Cycle-By-Cycle Duty Mode

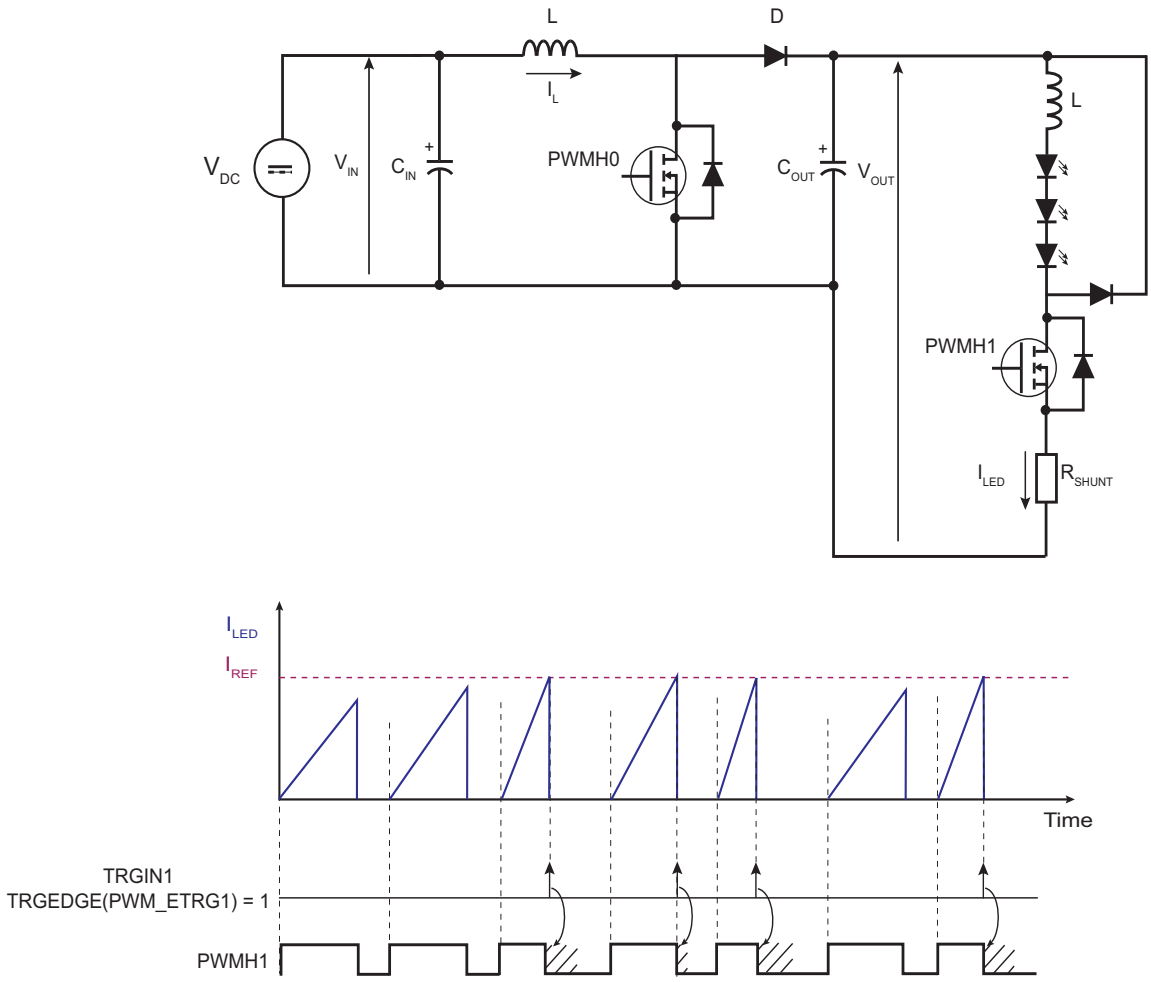


Application Example

Figure 53-32 illustrates an application example of the Cycle-by-cycle Duty mode.

In an LED string control circuit, Cycle-by-cycle Duty mode can be used to automatically limit the current in the LED string.

Figure 53-32. Cycle-By-Cycle Duty Mode: LED String Control



53.6.5.4 Leading-Edge Blanking (LEB)

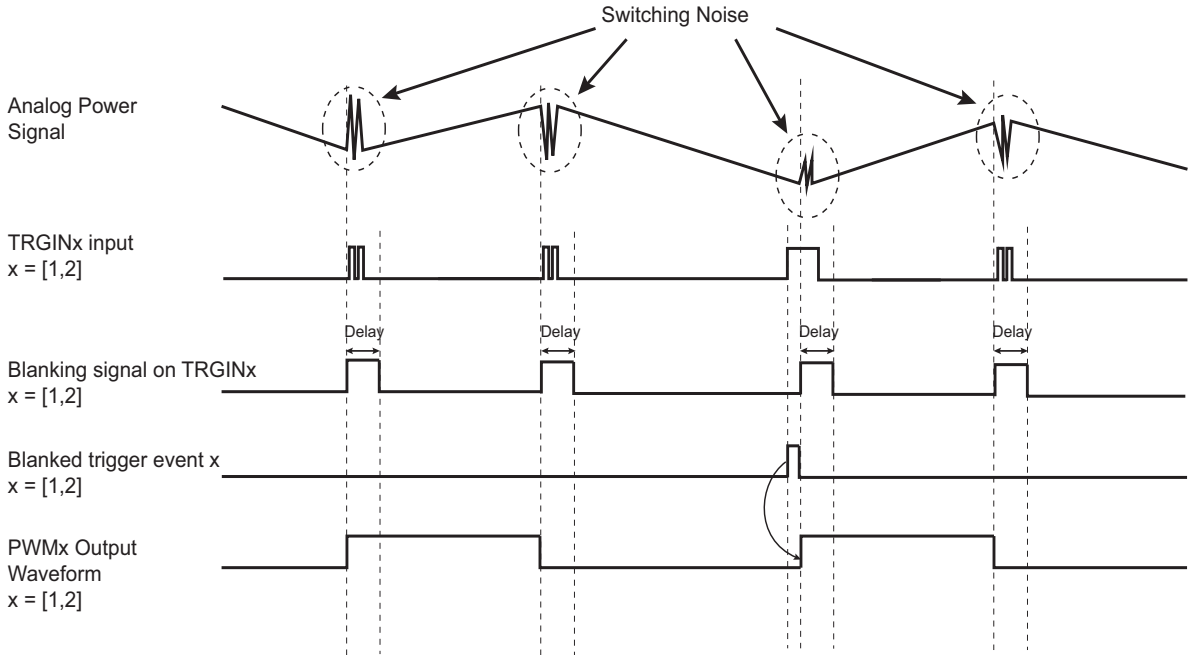
PWM channels 1 and 2 support leading-edge blanking. Leading-edge blanking masks the external trigger input when a transient occurs on the corresponding PWM output. It masks potential spurious external events due to power transistor switching.

The blanking delay on each external trigger input is configured by programming the LEBDELAYx in the PWM Leading-Edge Blanking Register.

The LEB can be enabled on both the rising and the falling edges for the PWMH and PWML outputs through the bits PWMLFEN, PWMLREN, PWMHFEN, PWMHREN.

Any event on the PWMEXTRGx input which occurs during the blanking time is ignored.

Figure 53-33. Leading-Edge Blanking



53.6.6 PWM Controller Operations

53.6.6.1 Initialization

Before enabling the channels, they must be configured by the software application as described below:

- Unlock User Interface by writing the WPCMD field in PWM_WPCR.
- Configuration of the clock generator (DIVA, PREA, DIVB, PREB in the PWM_CLK register if required).
- Selection of the clock for each channel (CPRE field in PWM_CMRx)
- Configuration of the waveform alignment for each channel (CALG field in PWM_CMRx)
- Selection of the counter event selection (if CALG = 1) for each channel (CES field in PWM_CMRx)
- Configuration of the output waveform polarity for each channel (CPOL bit in PWM_CMRx)
- Configuration of the period for each channel (CPRD in the PWM_CPRDx register). Writing in PWM_CPRDx register is possible while the channel is disabled. After validation of the channel, the user must use PWM_CPRDUPDx register to update PWM_CPRDx as explained below.
- Configuration of the duty-cycle for each channel (CDTY in the PWM_CDTYx register). Writing in PWM_CDTYx register is possible while the channel is disabled. After validation of the channel, the user must use PWM_CDTYUPDx register to update PWM_CDTYx as explained below.
- Configuration of the dead-time generator for each channel (DTH and DTL in PWM_DTx) if enabled (DTE bit in PWM_CMRx). Writing in the PWM_DTx register is possible while the channel is disabled. After validation of the channel, the user must use PWM_DTUPDx register to update PWM_DTx
- Selection of the synchronous channels (SYNCx in the PWM_SCM register)
- Selection of the moment when the WRDY flag and the corresponding DMA Controller transfer request are set (PTRM and PTRCS in the PWM_SCM register)
- Configuration of the Update mode (UPDM in PWM_SCM register)
- Configuration of the update period (UPR in PWM_SCUP register) if needed
- Configuration of the comparisons (PWM_CMPVx and PWM_CMPMx)
- Configuration of the event lines (PWM_ELMRx)
- Configuration of the fault inputs polarity (FPOL in PWM_FMR)
- Configuration of the fault protection (FMOD and FFIL in PWM_FMR, PWM_FPV and PWM_FPE1)
- Enable of the interrupts (writing CHIDx and FCHIDx in PWM_IER1, and writing WRDY, UNRE, CMPMx and CMPUx in PWM_IER2)
- Enable of the PWM channels (writing CHIDx in the PWM_ENA register)

53.6.6.2 Source Clock Selection Criteria

The large number of source clocks can make selection difficult. The relationship between the value in the PWM Channel Period Register (PWM_CPRDx) and the PWM Channel Duty Cycle Register (PWM_CDTYx) helps the user select the appropriate clock. The event number written in the Period Register gives the PWM accuracy. The Duty-Cycle quantum cannot be lower than $1/CPRDx$ value. The higher the value of PWM_CPRDx, the greater the PWM accuracy.

For example, if the user sets 15 (in decimal) in PWM_CPRDx, the user is able to set a value from between 1 up to 14 in PWM_CDTYx. The resulting duty-cycle quantum cannot be lower than 1/15 of the PWM period.

53.6.6.3 Changing the Duty-Cycle, the Period and the Dead-Times

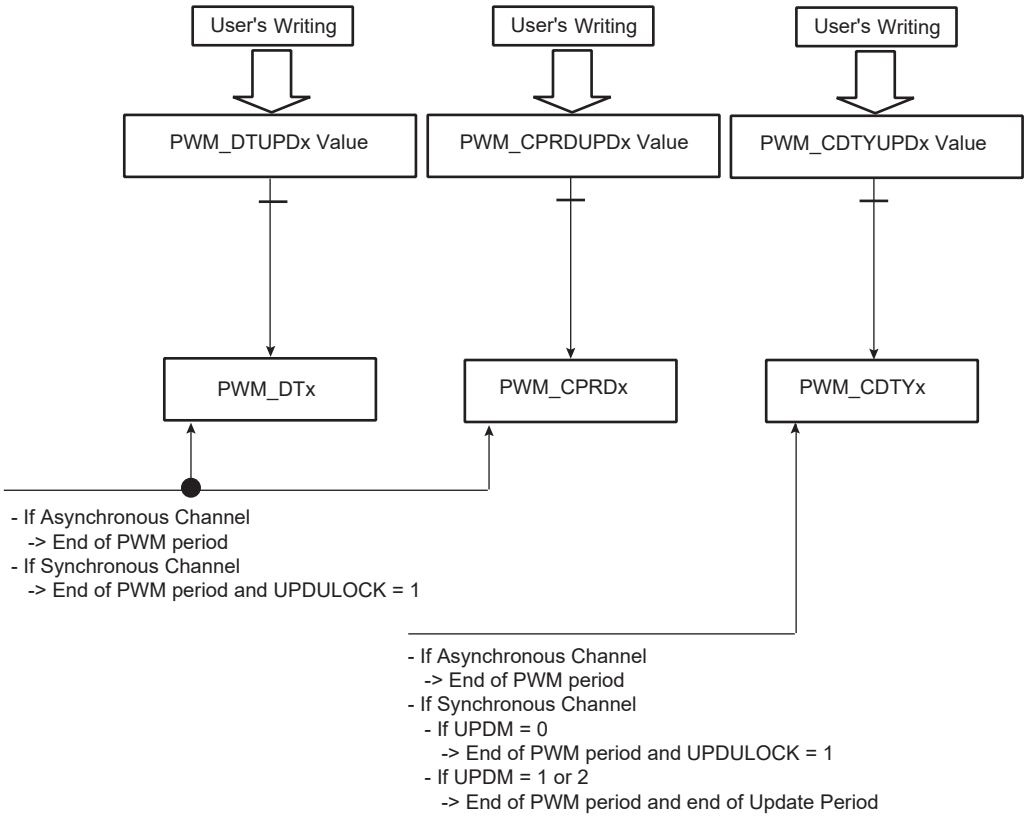
It is possible to modulate the output waveform duty-cycle, period and dead-times.

To prevent unexpected output waveform, the user must use the PWM Channel Duty Cycle Update Register (PWM_CDTYUPDx), the PWM Channel Period Update Register (PWM_CPRDUPDx) and the PWM Channel Dead Time Update Register (PWM_DTUPDx) to change waveform parameters while the channel is still enabled.

- If the channel is an asynchronous channel (SYNCx = 0 in PWM Sync Channels Mode Register (PWM_SCM)), these registers hold the new period, duty-cycle and dead-times values until the end of the current PWM period and update the values for the next period.
- If the channel is a synchronous channel and update method 0 is selected (SYNCx = 1 and UPDM = 0 in PWM_SCM register), these registers hold the new period, duty-cycle and dead-times values until the bit UPDULOCK is written at '1' (in PWM Sync Channels Update Control Register (PWM_SCUC)) and the end of the current PWM period, then update the values for the next period.
- If the channel is a synchronous channel and update method 1 or 2 is selected (SYNCx = 1 and UPDM = 1 or 2 in PWM_SCM register):
  - registers PWM_CPRDUPDx and PWM_DTUPDx hold the new period and dead-times values until the bit UPDULOCK is written at '1' (in PWM_SCUC) and the end of the current PWM period, then update the values for the next period.
  - register PWM_CDTYUPDx holds the new duty-cycle value until the end of the update period of synchronous channels (when UPRCNT is equal to UPR in PWM Sync Channels Update Period Register (PWM_SCUP)) and the end of the current PWM period, then updates the value for the next period.

Note: If the update registers PWM_CDTYUPDx, PWM_CPRDUPDx and PWM_DTUPDx are written several times between two updates, only the last written value is taken into account.

Figure 53-34. Synchronized Period, Duty-Cycle and Dead-Time Update



53.6.6.4 Changing the Update Period of Synchronous Channels

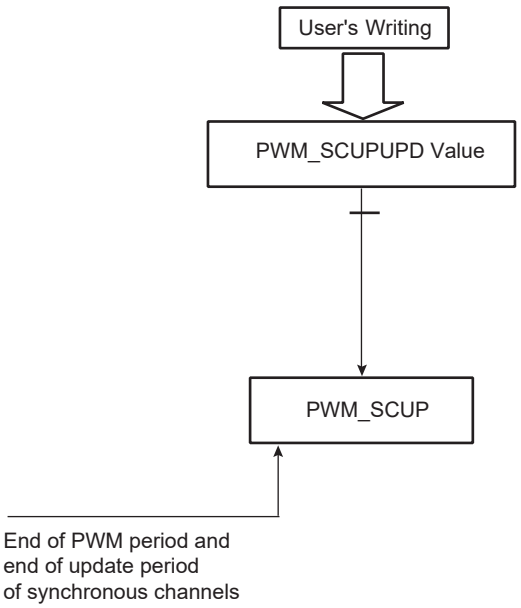
It is possible to change the update period of synchronous channels while they are enabled. See "Method 2: Manual write of duty-cycle values and automatic trigger of the update" and "Method 3: Automatic write of duty-cycle values and automatic trigger of the update" .

To prevent an unexpected update of the synchronous channels registers, the user must use the PWM Sync Channels Update Period Update Register (PWM_SCUPUPD) to change the update period of synchronous channels while they are still enabled. This register holds the new value until the end of the update period of synchronous channels (when UPRCNT is equal to UPR in PWM_SCUP) and the end of the current PWM period, then updates the value for the next period.

Note: If the update register PWM_SCUPUPD is written several times between two updates, only the last written value is taken into account.

Note: Changing the update period does make sense only if there is one or more synchronous channels and if the update method 1 or 2 is selected (UPDM = 1 or 2 in PWM Sync Channels Mode Register).

Figure 53-35. Synchronized Update of Update Period Value of Synchronous Channels



53.6.6.5 Changing the Comparison Value and the Comparison Configuration

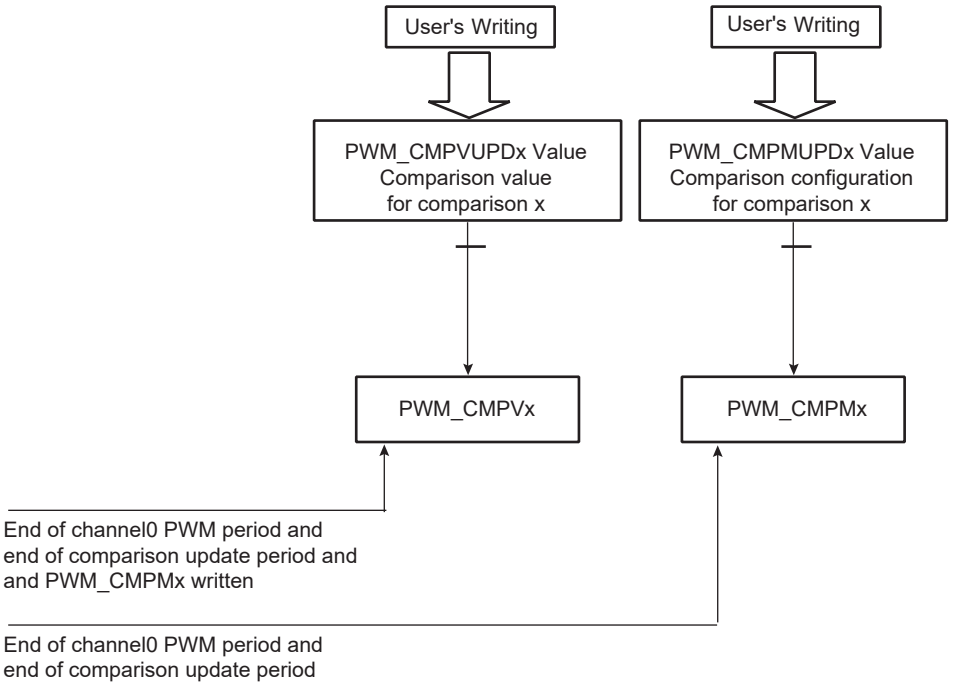
It is possible to change the comparison values and the comparison configurations while the channel 0 is enabled (see Section 53.6.3 "PWM Comparison Units").

To prevent unexpected comparison match, the user must use the PWM Comparison x Value Update Register (PWM_CMPVUPDx) and the PWM Comparison x Mode Update Register (PWM_CMPMUPDx) to change, respectively, the comparison values and the comparison configurations while the channel 0 is still enabled. These registers hold the new values until the end of the comparison update period (when CUPRCNT is equal to CUPR in PWM Comparison x Mode Register (PWM_CMPMx) and the end of the current PWM period, then update the values for the next period.

CAUTION: The write of the register PWM_CMPVUPDx must be followed by a write of the register PWM_CMPMUPDx.

Note: If the update registers PWM_CMPVUPDx and PWM_CMPMUPDx are written several times between two updates, only the last written value are taken into account.

Figure 53-36. Synchronized Update of Comparison Values and Configurations



53.6.6.6 Interrupt Sources

Depending on the interrupt mask in PWM_IMR1 and PWM_IMR2, an interrupt can be generated at the end of the corresponding channel period (CHIDx in the PWM Interrupt Status Register 1 (PWM_ISR1)), after a fault event (FCHIDx in PWM_ISR1), after a comparison match (CMPMx in PWM_ISR2), after a comparison update (CMPUx in PWM_ISR2) or according to the Transfer mode of the synchronous channels (WRDY and UNRE in PWM_ISR2).

If the interrupt is generated by the flags CHIDx or FCHIDx, the interrupt remains active until a read operation in PWM_ISR1 occurs.

If the interrupt is generated by the flags WRDY or UNRE or CMPMx or CMPUx, the interrupt remains active until a read operation in PWM_ISR2 occurs.

A channel interrupt is enabled by setting the corresponding bit in PWM_IER1 and PWM_IER2. A channel interrupt is disabled by setting the corresponding bit in PWM_IDR1 and PWM_IDR2.

53.6.7 Register Write Protection

To prevent any single software error that may corrupt PWM behavior, the registers listed below can be write-protected by writing the field WPCMD in the PWM Write Protection Control Register (PWM_WPCR). They are divided into six groups:

- Register group 0:
  - PWM Clock Register
- Register group 1:
  - PWM Disable Register
- Register group 2:
  - PWM Sync Channels Mode Register
  - PWM Channel Mode Register
  - PWM Stepper Motor Mode Register
  - PWM Fault Protection Value Register 2
  - PWM Leading-Edge Blanking Register
  - PWM Channel Mode Update Register
- Register group 3:
  - PWM Spread Spectrum Register
  - PWM Spread Spectrum Update Register
  - PWM Channel Period Register
  - PWM Channel Period Update Register
- Register group 4:
  - PWM Channel Dead Time Register
  - PWM Channel Dead Time Update Register
- Register group 5:
  - PWM Fault Mode Register
  - PWM Fault Protection Value Register 1

There are two types of write protection:

- SW write protection—can be enabled or disabled by software
- HW write protection—can be enabled by software but only disabled by a hardware reset of the PWM controller

Both types of write protection can be applied independently to a particular register group by means of the WPCMD and WPRGx fields in PWM_WPCR. If at least one type of write protection is active, the register group is write-protected. The value of field WPCMD defines the action to be performed:

- 0: Disables SW write protection of the register groups of which the bit WPRGx is at ‘1’
- 1: Enables SW write protection of the register groups of which the bit WPRGx is at ‘1’
- 2: Enables HW write protection of the register groups of which the bit WPRGx is at ‘1’

At any time, the user can determine whether SW or HW write protection is active in a particular register group by the fields WPSWS and WPHWS in the PWM Write Protection Status Register (PWM_WPSR).

If a write access to a write-protected register is detected, the WPVS flag in PWM_WPSR is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS and WPVSRC fields are automatically cleared after reading PWM_WPSR.

53.7 Pulse Width Modulation Controller (PWM) User Interface

Table 53-8. Register Mapping

Offset	Register	Name	Access	Reset
0x00	PWM Clock Register	PWM_CLK	Read/Write	0x0
0x04	PWM Enable Register	PWM_ENA	Write-only	–
0x08	PWM Disable Register	PWM_DIS	Write-only	–
0x0C	PWM Status Register	PWM_SR	Read-only	0x0
0x10	PWM Interrupt Enable Register 1	PWM_IER1	Write-only	–
0x14	PWM Interrupt Disable Register 1	PWM_IDR1	Write-only	–
0x18	PWM Interrupt Mask Register 1	PWM_IMR1	Read-only	0x0
0x1C	PWM Interrupt Status Register 1	PWM_ISR1	Read-only	0x0
0x20	PWM Sync Channels Mode Register	PWM_SCM	Read/Write	0x0
0x24	PWM DMA Register	PWM_DMAR	Write-only	–
0x28	PWM Sync Channels Update Control Register	PWM_SCUC	Read/Write	0x0
0x2C	PWM Sync Channels Update Period Register	PWM_SCUP	Read/Write	0x0
0x30	PWM Sync Channels Update Period Update Register	PWM_SCUPUPD	Write-only	–
0x34	PWM Interrupt Enable Register 2	PWM_IER2	Write-only	–
0x38	PWM Interrupt Disable Register 2	PWM_IDR2	Write-only	–
0x3C	PWM Interrupt Mask Register 2	PWM_IMR2	Read-only	0x0
0x40	PWM Interrupt Status Register 2	PWM_ISR2	Read-only	0x0
0x44	PWM Output Override Value Register	PWM_OOV	Read/Write	0x0
0x48	PWM Output Selection Register	PWM_OS	Read/Write	0x0
0x4C	PWM Output Selection Set Register	PWM_OSS	Write-only	–
0x50	PWM Output Selection Clear Register	PWM_OSC	Write-only	–
0x54	PWM Output Selection Set Update Register	PWM_OSSUPD	Write-only	–
0x58	PWM Output Selection Clear Update Register	PWM_OSCUPD	Write-only	–
0x5C	PWM Fault Mode Register	PWM_FMR	Read/Write	0x0
0x60	PWM Fault Status Register	PWM_FSR	Read-only	0x0
0x64	PWM Fault Clear Register	PWM_FCR	Write-only	–
0x68	PWM Fault Protection Value Register 1	PWM_FPV1	Read/Write	0x0
0x6C	PWM Fault Protection Enable Register	PWM_FPE	Read/Write	0x0
0x70–0x78	Reserved	–	–	–
0x7C	PWM Event Line 0 Mode Register	PWM_ELMR0	Read/Write	0x0
0x80	PWM Event Line 1 Mode Register	PWM_ELMR1	Read/Write	0x0
0x84–0x9C	Reserved	–	–	–
0xA0	PWM Spread Spectrum Register	PWM_SSPR	Read/Write	0x0
0xA4	PWM Spread Spectrum Update Register	PWM_SSPUP	Write-only	–
0xA8–0xAC	Reserved	–	–	–

Table 53-8. Register Mapping (Continued)

Offset	Register	Name	Access	Reset
0xB0	PWM Stepper Motor Mode Register	PWM_SMMR	Read/Write	0x0
0xC0	PWM Fault Protection Value 2 Register	PWM_FPV2	Read/Write	0x003F_003F
0xC4–0xE0	Reserved	–	–	–
0xE4	PWM Write Protection Control Register	PWM_WPCR	Write-only	–
0xE8	PWM Write Protection Status Register	PWM_WPSR	Read-only	0x0
0xEC–0xFC	Reserved	–	–	–
0x100–0x12C	Reserved	–	–	–
0x130	PWM Comparison 0 Value Register	PWM_CMPV0	Read/Write	0x0
0x134	PWM Comparison 0 Value Update Register	PWM_CMPVUPD0	Write-only	–
0x138	PWM Comparison 0 Mode Register	PWM_CMPM0	Read/Write	0x0
0x13C	PWM Comparison 0 Mode Update Register	PWM_CMPMUPD0	Write-only	–
0x140	PWM Comparison 1 Value Register	PWM_CMPV1	Read/Write	0x0
0x144	PWM Comparison 1 Value Update Register	PWM_CMPVUPD1	Write-only	–
0x148	PWM Comparison 1 Mode Register	PWM_CMPM1	Read/Write	0x0
0x14C	PWM Comparison 1 Mode Update Register	PWM_CMPMUPD1	Write-only	–
0x150	PWM Comparison 2 Value Register	PWM_CMPV2	Read/Write	0x0
0x154	PWM Comparison 2 Value Update Register	PWM_CMPVUPD2	Write-only	–
0x158	PWM Comparison 2 Mode Register	PWM_CMPM2	Read/Write	0x0
0x15C	PWM Comparison 2 Mode Update Register	PWM_CMPMUPD2	Write-only	–
0x160	PWM Comparison 3 Value Register	PWM_CMPV3	Read/Write	0x0
0x164	PWM Comparison 3 Value Update Register	PWM_CMPVUPD3	Write-only	–
0x168	PWM Comparison 3 Mode Register	PWM_CMPM3	Read/Write	0x0
0x16C	PWM Comparison 3 Mode Update Register	PWM_CMPMUPD3	Write-only	–
0x170	PWM Comparison 4 Value Register	PWM_CMPV4	Read/Write	0x0
0x174	PWM Comparison 4 Value Update Register	PWM_CMPVUPD4	Write-only	–
0x178	PWM Comparison 4 Mode Register	PWM_CMPM4	Read/Write	0x0
0x17C	PWM Comparison 4 Mode Update Register	PWM_CMPMUPD4	Write-only	–
0x180	PWM Comparison 5 Value Register	PWM_CMPV5	Read/Write	0x0
0x184	PWM Comparison 5 Value Update Register	PWM_CMPVUPD5	Write-only	–
0x188	PWM Comparison 5 Mode Register	PWM_CMPM5	Read/Write	0x0
0x18C	PWM Comparison 5 Mode Update Register	PWM_CMPMUPD5	Write-only	–
0x190	PWM Comparison 6 Value Register	PWM_CMPV6	Read/Write	0x0
0x194	PWM Comparison 6 Value Update Register	PWM_CMPVUPD6	Write-only	–
0x198	PWM Comparison 6 Mode Register	PWM_CMPM6	Read/Write	0x0
0x19C	PWM Comparison 6 Mode Update Register	PWM_CMPMUPD6	Write-only	–
0x1A0	PWM Comparison 7 Value Register	PWM_CMPV7	Read/Write	0x0
0x1A4	PWM Comparison 7 Value Update Register	PWM_CMPVUPD7	Write-only	–

Table 53-8. Register Mapping (Continued)

Offset	Register	Name	Access	Reset
0x1A8	PWM Comparison 7 Mode Register	PWM_CMPM7	Read/Write	0x0
0x1AC	PWM Comparison 7 Mode Update Register	PWM_CMPMUPD7	Write-only	–
0x1B0–0x1FC	Reserved	–	–	–
0x200 + ch_num * 0x20 + 0x00	PWM Channel Mode Register(1)	PWM_CMR	Read/Write	0x0
0x200 + ch_num * 0x20 + 0x04	PWM Channel Duty Cycle Register(1)	PWM_CDTY	Read/Write	0x0
0x200 + ch_num * 0x20 + 0x08	PWM Channel Duty Cycle Update Register(1)	PWM_CDTYUPD	Write-only	–
0x200 + ch_num * 0x20 + 0x0C	PWM Channel Period Register(1)	PWM_CPRD	Read/Write	0x0
0x200 + ch_num * 0x20 + 0x10	PWM Channel Period Update Register(1)	PWM_CPRDUPD	Write-only	–
0x200 + ch_num * 0x20 + 0x14	PWM Channel Counter Register(1)	PWM_CCNT	Read-only	0x0
0x200 + ch_num * 0x20 + 0x18	PWM Channel Dead Time Register(1)	PWM_DT	Read/Write	0x0
0x200 + ch_num * 0x20 + 0x1C	PWM Channel Dead Time Update Register(1)	PWM_DTUPD	Write-only	–
0x400 + ch_num * 0x20 + 0x00	PWM Channel Mode Update Register(1)	PWM_CMUPD	Write-only	–
0x400 + trg_num * 0x20 + 0x0C	PWM External Trigger Register(2)	PWM_ETRG	Read/Write	0x0
0x400 + trg_num * 0x20 + 0x10	PWM Leading-Edge Blanking Register(2)	PWM_LEBR	Read/Write	0x0

Notes: 1. Some registers are indexed with “ch_num” index ranging from 0 to 3.
2. Some registers are indexed with “trg_num” index ranging from 1 to 2.

53.7.1 PWM Clock Register

Name: PWM_CLK

Address: 0xF802C000

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	PREB			
23	22	21	20	19	18	17	16
DIVB							
15	14	13	12	11	10	9	8
–	–	–	–	PREA			
7	6	5	4	3	2	1	0
DIVA							

This register can only be written if bits WPSWS0 and WPHWS0 are cleared in the PWM Write Protection Status Register.

- **DIVA: CLKA Divide Factor**

Value	Name	Description
0	CLKA_POFF	CLKA clock is turned off
1	PREA	CLKA clock is clock selected by PREA
2–255	PREA_DIV	CLKA clock is clock selected by PREA divided by DIVA factor

- **DIVB: CLKB Divide Factor**

Value	Name	Description
0	CLKB_POFF	CLKB clock is turned off
1	PREB	CLKB clock is clock selected by PREB
2–255	PREB_DIV	CLKB clock is clock selected by PREB divided by DIVB factor

- **PREA: CLKA Source Clock Selection**

Value	Name	Description
0	CLK	Peripheral clock
1	CLK_DIV2	Peripheral clock/2
2	CLK_DIV4	Peripheral clock/4
3	CLK_DIV8	Peripheral clock/8
4	CLK_DIV16	Peripheral clock/16
5	CLK_DIV32	Peripheral clock/32
6	CLK_DIV64	Peripheral clock/64
7	CLK_DIV128	Peripheral clock/128
8	CLK_DIV256	Peripheral clock/256
9	CLK_DIV512	Peripheral clock/512
10	CLK_DIV1024	Peripheral clock/1024
Other	–	Reserved

- **PREB: CLKB Source Clock Selection**

Value	Name	Description
0	CLK	Peripheral clock
1	CLK_DIV2	Peripheral clock/2
2	CLK_DIV4	Peripheral clock/4
3	CLK_DIV8	Peripheral clock/8
4	CLK_DIV16	Peripheral clock/16
5	CLK_DIV32	Peripheral clock/32
6	CLK_DIV64	Peripheral clock/64
7	CLK_DIV128	Peripheral clock/128
8	CLK_DIV256	Peripheral clock/256
9	CLK_DIV512	Peripheral clock/512
10	CLK_DIV1024	Peripheral clock/1024
Other	–	Reserved

53.7.2 PWM Enable Register

Name: PWM_ENA

Address: 0xF802C004

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	CHID3	CHID2	CHID1	CHID0

- **CHIDx: Channel ID**

0: No effect.

1: Enable PWM output for channel x.

53.7.3 PWM Disable Register

Name: PWM_DIS

Address: 0xF802C008

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	CHID3	CHID2	CHID1	CHID0

This register can only be written if bits WPSWS1 and WPHWS1 are cleared in the PWM Write Protection Status Register.

- **CHIDx: Channel ID**

0: No effect.

1: Disable PWM output for channel x.

53.7.4 PWM Status Register

Name: PWM_SR

Address: 0xF802C00C

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	CHID3	CHID2	CHID1	CHID0

- **CHIDx: Channel ID**

0: PWM output for channel x is disabled.

1: PWM output for channel x is enabled.

53.7.5 PWM Interrupt Enable Register 1

Name: PWM_IER1

Address: 0xF802C010

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	FCHID3	FCHID2	FCHID1	FCHID0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	CHID3	CHID2	CHID1	CHID0

- **CHIDx: Counter Event on Channel x Interrupt Enable**
- **FCHIDx: Fault Protection Trigger on Channel x Interrupt Enable**

53.7.6 PWM Interrupt Disable Register 1

Name: PWM_IDR1

Address: 0xF802C014

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	FCHID3	FCHID2	FCHID1	FCHID0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	CHID3	CHID2	CHID1	CHID0

- **CHIDx: Counter Event on Channel x Interrupt Disable**
- **FCHIDx: Fault Protection Trigger on Channel x Interrupt Disable**

53.7.7 PWM Interrupt Mask Register 1

Name: PWM_IMR1

Address: 0xF802C018

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	FCHID3	FCHID2	FCHID1	FCHID0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	CHID3	CHID2	CHID1	CHID0

- **CHIDx: Counter Event on Channel x Interrupt Mask**
- **FCHIDx: Fault Protection Trigger on Channel x Interrupt Mask**

53.7.8 PWM Interrupt Status Register 1

Name: PWM_ISR1

Address: 0xF802C01C

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	FCHID3	FCHID2	FCHID1	FCHID0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	CHID3	CHID2	CHID1	CHID0

- **CHIDx: Counter Event on Channel x**

0: No new counter event has occurred since the last read of PWM_ISR1.

1: At least one counter event has occurred since the last read of PWM_ISR1.

- **FCHIDx: Fault Protection Trigger on Channel x**

0: No new trigger of the fault protection since the last read of PWM_ISR1.

1: At least one trigger of the fault protection since the last read of PWM_ISR1.

Note: Reading PWM_ISR1 automatically clears CHIDx and FCHIDx flags.

53.7.9 PWM Sync Channels Mode Register

Name: PWM_SCM

Address: 0xF802C020

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
PTRCS			PTRM	–	–	UPDM	
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	SYNC3	SYNC2	SYNC1	SYNC0

This register can only be written if bits WPSWS2 and WPHWS2 are cleared in the PWM Write Protection Status Register.

- **SYNCx: Synchronous Channel x**

0: Channel x is not a synchronous channel.

1: Channel x is a synchronous channel.

- **UPDM: Synchronous Channels Update Mode**

Value	Name	Description
0	MODE0	Manual write of double buffer registers and manual update of synchronous channels(1)
1	MODE1	Manual write of double buffer registers and automatic update of synchronous channels(2)
2	MODE2	Automatic write of duty-cycle update registers by the DMA Controller and automatic update of synchronous channels(2)

Notes: 1. The update occurs at the beginning of the next PWM period, when the UPDULOCK bit in PWM Sync Channels Update Control Register is set.

2. The update occurs when the Update Period is elapsed.

- **PTRM: DMA Controller Transfer Request Mode**

UPDM	PTRM	WRDY Flag and DMA Controller Transfer Request
0	x	The WRDY flag in PWM Interrupt Status Register 2 and the DMA transfer request are never set to ‘1’.
1	x	The WRDY flag in PWM Interrupt Status Register 2 is set to ‘1’ as soon as the update period is elapsed, the DMA Controller transfer request is never set to ‘1’.
2	0	The WRDY flag in PWM Interrupt Status Register 2 and the DMA transfer request are set to ‘1’ as soon as the update period is elapsed.
	1	The WRDY flag in PWM Interrupt Status Register 2 and the DMA transfer request are set to ‘1’ as soon as the selected comparison matches.

- **PTRCS: DMA Controller Transfer Request Comparison Selection**

Selection of the comparison used to set the flag WRDY and the corresponding DMA Controller transfer request.

53.7.10 PWM DMA Register

Name: PWM_DMAR

Address: 0xF802C024

Access: Write- only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
DMADUTY							

15	14	13	12	11	10	9	8
DMADUTY							

7	6	5	4	3	2	1	0
DMADUTY							

Only the first 16 bits (channel counter size) are significant.

- **DMADUTY: Duty-Cycle Holding Register for DMA Access**

Each write access to PWM_DMAR sequentially updates the CDTY field of PWM_CDTYx with DMADUTY (only for channel configured as synchronous). See “Method 3: Automatic write of duty-cycle values and automatic trigger of the update” .

53.7.11 PWM Sync Channels Update Control Register

Name: PWM_SCUC

Address: 0xF802C028

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	UPDULOCK

- **UPDULOCK: Synchronous Channels Update Unlock**

0: No effect

1: If the UPDM field is set to ‘0’ in PWM Sync Channels Mode Register, writing the UPDULOCK bit to ‘1’ triggers the update of the period value, the duty-cycle and the dead-time values of synchronous channels at the beginning of the next PWM period. If the field UPDM is set to ‘1’ or ‘2’, writing the UPDULOCK bit to ‘1’ triggers only the update of the period value and of the dead-time values of synchronous channels.

This bit is automatically reset when the update is done.

53.7.12 PWM Sync Channels Update Period Register

Name: PWM_SCUP

Address: 0xF802C02C

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
UPRCNT				UPR			

- **UPR: Update Period**

Defines the time between each update of the synchronous channels if automatic trigger of the update is activated (UPDM = 1 or UPDM = 2 in PWM Sync Channels Mode Register). This time is equal to UPR+1 periods of the synchronous channels.

- **UPRCNT: Update Period Counter**

Reports the value of the update period counter.

53.7.13 PWM Sync Channels Update Period Update Register

Name: PWM_SCUPUPD

Address: 0xF802C030

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	UPRUPD			

This register acts as a double buffer for the UPR value. This prevents an unexpected automatic trigger of the update of synchronous channels.

- **UPRUPD: Update Period Update**

Defines the wanted time between each update of the synchronous channels if automatic trigger of the update is activated (UPDM = 1 or UPDM = 2 in PWM Sync Channels Mode Register). This time is equal to UPR+1 periods of the synchronous channels.

53.7.14 PWM Interrupt Enable Register 2

Name: PWM_IER2

Address: 0xF802C034

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
15	14	13	12	11	10	9	8
CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
7	6	5	4	3	2	1	0
–	–	–	–	UNRE	–	–	WRDY

- **WRDY: Write Ready for Synchronous Channels Update Interrupt Enable**
- **UNRE: Synchronous Channels Update Underrun Error Interrupt Enable**
- **CMPMx: Comparison x Match Interrupt Enable**
- **CMPUx: Comparison x Update Interrupt Enable**

53.7.15 PWM Interrupt Disable Register 2

Name: PWM_IDR2

Address: 0xF802C038

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
15	14	13	12	11	10	9	8
CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
7	6	5	4	3	2	1	0
–	–	–	–	UNRE	–	–	WRDY

- **WRDY: Write Ready for Synchronous Channels Update Interrupt Disable**
- **UNRE: Synchronous Channels Update Underrun Error Interrupt Disable**
- **CMPMx: Comparison x Match Interrupt Disable**
- **CMPUx: Comparison x Update Interrupt Disable**

53.7.16 PWM Interrupt Mask Register 2

Name: PWM_IMR2

Address: 0xF802C03C

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
15	14	13	12	11	10	9	8
CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
7	6	5	4	3	2	1	0
–	–	–	–	UNRE	–	–	WRDY

- **WRDY: Write Ready for Synchronous Channels Update Interrupt Mask**
- **UNRE: Synchronous Channels Update Underrun Error Interrupt Mask**
- **CMPMx: Comparison x Match Interrupt Mask**
- **CMPUx: Comparison x Update Interrupt Mask**

53.7.17 PWM Interrupt Status Register 2

Name: PWM_ISR2

Address: 0xF802C040

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
15	14	13	12	11	10	9	8
CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
7	6	5	4	3	2	1	0
–	–	–	–	UNRE	–	–	WRDY

- **WRDY: Write Ready for Synchronous Channels Update**

0: New duty-cycle and dead-time values for the synchronous channels cannot be written.

1: New duty-cycle and dead-time values for the synchronous channels can be written.

- **UNRE: Synchronous Channels Update Underrun Error**

0: No Synchronous Channels Update Underrun has occurred since the last read of the PWM_ISR2 register.

1: At least one Synchronous Channels Update Underrun has occurred since the last read of the PWM_ISR2 register.

- **CMPMx: Comparison x Match**

0: The comparison x has not matched since the last read of the PWM_ISR2 register.

1: The comparison x has matched at least one time since the last read of the PWM_ISR2 register.

- **CMPUx: Comparison x Update**

0: The comparison x has not been updated since the last read of the PWM_ISR2 register.

1: The comparison x has been updated at least one time since the last read of the PWM_ISR2 register.

Note: Reading PWM_ISR2 automatically clears flags WRDY, UNRE and CMPSx.

53.7.18 PWM Output Override Value Register

Name: PWM_OOV

Address: 0xF802C044

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	OOVL3	OOVL2	OOVL1	OOVL0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	OOVH3	OOVH2	OOVH1	OOVH0

- **OOVHx: Output Override Value for PWMH output of the channel x**

0: Override value is 0 for PWMH output of channel x.

1: Override value is 1 for PWMH output of channel x.

- **OOVLx: Output Override Value for PWML output of the channel x**

0: Override value is 0 for PWML output of channel x.

1: Override value is 1 for PWML output of channel x.

53.7.19 PWM Output Selection Register

Name: PWM_OS

Address: 0xF802C048

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	OSL3	OSL2	OSL1	OSL0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	OSH3	OSH2	OSH1	OSH0

- **OSHx: Output Selection for PWMH output of the channel x**

0: Dead-time generator output DTOHx selected as PWMH output of channel x.

1: Output override value OOVHx selected as PWMH output of channel x.

- **OSLx: Output Selection for PWML output of the channel x**

0: Dead-time generator output DTOLx selected as PWML output of channel x.

1: Output override value OOVLx selected as PWML output of channel x.

53.7.20 PWM Output Selection Set Register

Name: PWM_OSS

Address: 0xF802C04C

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	OSSL3	OSSL2	OSSL1	OSSL0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	OSSH3	OSSH2	OSSH1	OSSH0

- **OSSHx: Output Selection Set for PWMH output of the channel x**

0: No effect.

1: Output override value OOVHx selected as PWMH output of channel x.

- **OSSLx: Output Selection Set for PWML output of the channel x**

0: No effect.

1: Output override value OOVLx selected as PWML output of channel x.

53.7.21 PWM Output Selection Clear Register

Name: PWM_OSC

Address: 0xF802C050

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	OSCL3	OSCL2	OSCL1	OSCL0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	OSCH3	OSCH2	OSCH1	OSCH0

- **OSCHx: Output Selection Clear for PWMH output of the channel x**

0: No effect.

1: Dead-time generator output DTOHx selected as PWMH output of channel x.

- **OSCLx: Output Selection Clear for PWML output of the channel x**

0: No effect.

1: Dead-time generator output DTOLx selected as PWML output of channel x.

53.7.22 PWM Output Selection Set Update Register

Name: PWM_OSSUPD

Address: 0xF802C054

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	OSSUPL3	OSSUPL2	OSSUPL1	OSSUPL0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	OSSUPH3	OSSUPH2	OSSUPH1	OSSUPH0

- **OSSUPHx: Output Selection Set for PWMH output of the channel x**

0: No effect.

1: Output override value OOVHx selected as PWMH output of channel x at the beginning of the next channel x PWM period.

- **OSSUPLx: Output Selection Set for PWML output of the channel x**

0: No effect.

1: Output override value OOVLx selected as PWML output of channel x at the beginning of the next channel x PWM period.

53.7.23 PWM Output Selection Clear Update Register

Name: PWM_OSCUPD

Address: 0xF802C058

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	OSCUPL3	OSCUPL2	OSCUPL1	OSCUPL0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	OSCUPH3	OSCUPH2	OSCUPH1	OSCUPH0

- **OSCUPHx: Output Selection Clear for PWMH output of the channel x**

0: No effect.

1: Dead-time generator output DTOHx selected as PWMH output of channel x at the beginning of the next channel x PWM period.

- **OSCUPLx: Output Selection Clear for PWML output of the channel x**

0: No effect.

1: Dead-time generator output DTOLx selected as PWML output of channel x at the beginning of the next channel x PWM period.

53.7.24 PWM Fault Mode Register

Name: PWM_FMR

Address: 0xF802C05C

Access: Read/Write

31	30	29	28	27	26	25	24
–							

23	22	21	20	19	18	17	16
FFIL							

15	14	13	12	11	10	9	8
FMOD							

7	6	5	4	3	2	1	0
FPOL							

This register can only be written if bits WPSWS5 and WPHWS5 are cleared in the PWM Write Protection Status Register.

Refer to Section 53.5.4 “Fault Inputs” for details on fault generation.

- **FPOL: Fault Polarity**

For each bit y of FPOL, where y is the fault input number:

0: The fault y becomes active when the fault input y is at 0.

1: The fault y becomes active when the fault input y is at 1.

- **FMOD: Fault Activation Mode**

For each bit y of FMOD, where y is the fault input number:

0: The fault y is active until the fault condition is removed at the peripheral(1) level.

1: The fault y stays active until the fault condition is removed at the peripheral(1) level AND until it is cleared in the PWM Fault Clear Register.

Note: 1. The peripheral generating the fault.

- **FFIL: Fault Filtering**

For each bit y of FFIL, where y is the fault input number:

0: The fault input y is not filtered.

1: The fault input y is filtered.

CAUTION: To prevent an unexpected activation of the status flag FSy in the PWM Fault Status Register, the bit FMODy can be set to ‘1’ only if the FPOLy bit has been previously configured to its final value.

53.7.25 PWM Fault Status Register

Name: PWM_FSR

Address: 0xF802C060

Access: Read-only

31	30	29	28	27	26	25	24
–							
23	22	21	20	19	18	17	16
–							
15	14	13	12	11	10	9	8
FS							
7	6	5	4	3	2	1	0
FIV							

Refer to Section 53.5.4 “Fault Inputs” for details on fault generation.

- **FIV: Fault Input Value**

For each bit y of FIV, where y is the fault input number:

0: The current sampled value of the fault input y is 0 (after filtering if enabled).

1: The current sampled value of the fault input y is 1 (after filtering if enabled).

- **FS: Fault Status**

For each bit y of FS, where y is the fault input number:

0: The fault y is not currently active.

1: The fault y is currently active.

53.7.26 PWM Fault Clear Register

Name: PWM_FCR

Address: 0xF802C064

Access: Write-only

31	30	29	28	27	26	25	24
–							

23	22	21	20	19	18	17	16
–							

15	14	13	12	11	10	9	8
–							

7	6	5	4	3	2	1	0
FCLR							

Refer to Section 53.5.4 “Fault Inputs” for details on fault generation.

- **FCLR: Fault Clear**

For each bit y of FCLR, where y is the fault input number:

0: No effect.

1: If bit y of FMOD field is set to ‘1’ and if the fault input y is not at the level defined by the bit y of FPOL field, the fault y is cleared and becomes inactive (FMOD and FPOL fields belong to PWM Fault Mode Register), else writing this bit to ‘1’ has no effect.

53.7.27 PWM Fault Protection Value Register 1

Name: PWM_FPV1

Address: 0xF802C068

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	FPVL3	FPVL2	FPVL1	FPVL0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	FPVH3	FPVH2	FPVH1	FPVH0

This register can only be written if bits WPSWS5 and WPHWS5 are cleared in the PWM Write Protection Status Register.

Refer to Section 53.5.4 “Fault Inputs” for details on fault generation.

- **FPVHx: Fault Protection Value for PWMH output on channel x**

This bit is taken into account only if the bit FPZHx is set to ‘0’ in PWM Fault Protection Value Register 2.

0: PWMH output of channel x is forced to ‘0’ when fault occurs.

1: PWMH output of channel x is forced to ‘1’ when fault occurs.

- **FPVLx: Fault Protection Value for PWML output on channel x**

This bit is taken into account only if the bit FPZLx is set to ‘0’ in PWM Fault Protection Value Register 2.

0: PWML output of channel x is forced to ‘0’ when fault occurs.

1: PWML output of channel x is forced to ‘1’ when fault occurs.

53.7.28 PWM Fault Protection Enable Register

Name: PWM_FPE

Address: 0xF802C06C

Access: Read/Write

31	30	29	28	27	26	25	24
FPE3							
23	22	21	20	19	18	17	16
FPE2							
15	14	13	12	11	10	9	8
FPE1							
7	6	5	4	3	2	1	0
FPE0							

This register can only be written if bits WPSWS5 and WPHWS5 are cleared in the PWM Write Protection Status Register.

Only the first 6 bits (number of fault input pins) of fields FPE0, FPE1, FPE2 and FPE3 are significant.

Refer to Section 53.5.4 “Fault Inputs” for details on fault generation.

- **FPEx: Fault Protection Enable for channel x**

For each bit y of FPEx, where y is the fault input number:

0: Fault y is not used for the fault protection of channel x.

1: Fault y is used for the fault protection of channel x.

CAUTION: To prevent an unexpected activation of the fault protection, the bit y of FPEx field can be set to ‘1’ only if the corresponding FPOL field has been previously configured to its final value in PWM Fault Mode Register.

53.7.29 PWM Event Line x Register

Name: PWM_ELMRx

Address: 0xF802C07C

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
CSEL7	CSEL6	CSEL5	CSEL4	CSEL3	CSEL2	CSEL1	CSEL0

- **CSELy: Comparison y Selection**

0: A pulse is not generated on the event line x when the comparison y matches.

1: A pulse is generated on the event line x when the comparison y match.

53.7.30 PWM Spread Spectrum Register

Name: PWM_SSPR

Address: 0xF802C0A0

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	SPRDM
23	22	21	20	19	18	17	16
SPRD							
15	14	13	12	11	10	9	8
SPRD							
7	6	5	4	3	2	1	0
SPRD							

This register can only be written if bits WPSWS3 and WPHWS3 are cleared in the PWM Write Protection Status Register.

Only the first 16 bits (channel counter size) are significant.

- **SPRD: Spread Spectrum Limit Value**

The spread spectrum limit value defines the range for the spread spectrum counter. It is introduced in order to achieve constant varying PWM period for the output waveform.

- **SPRDM: Spread Spectrum Counter Mode**

0: Triangular mode. The spread spectrum counter starts to count from -SPRD when the channel 0 is enabled and counts upwards at each PWM period. When it reaches +SPRD, it restarts to count from -SPRD again.

1: Random mode. The spread spectrum counter is loaded with a new random value at each PWM period. This random value is uniformly distributed and is between -SPRD and +SPRD.

53.7.31 PWM Spread Spectrum Update Register

Name: PWM_SSPUP

Address: 0xF802C0A4

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
SPRDUP							
15	14	13	12	11	10	9	8
SPRDUP							
7	6	5	4	3	2	1	0
SPRDUP							

This register can only be written if bits WPSWS3 and WPHWS3 are cleared in the PWM Write Protection Status Register.

This register acts as a double buffer for the SPRD value. This prevents an unexpected waveform when modifying the spread spectrum limit value.

Only the first 16 bits (channel counter size) are significant.

- **SPRDUP: Spread Spectrum Limit Value Update**

The spread spectrum limit value defines the range for the spread spectrum counter. It is introduced in order to achieve constant varying period for the output waveform.

53.7.32 PWM Stepper Motor Mode Register

Name: PWM_SMMR

Address: 0xF802C0B0

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	DOWN1	DOWN0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	GCEN1	GCEN0

- **GCENx: Gray Count Enable**

0: Disable gray count generation on PWML[2*x], PWMH[2*x], PWML[2*x +1], PWMH[2*x +1]

1: Enable gray count generation on PWML[2*x], PWMH[2*x], PWML[2*x +1], PWMH[2*x +1].

- **DOWNx: Down Count**

0: Up counter.

1: Down counter.

53.7.33 PWM Fault Protection Value Register 2

Name: PWM_FPV2

Address: 0xF802C0C0

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	FPZL3	FPZL2	FPZL1	FPZL0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	FPZH3	FPZH2	FPZH1	FPZH0

This register can only be written if bits WPSWS5 and WPHWS5 are cleared in the PWM Write Protection Status Register.

- **FPZHx: Fault Protection to Hi-Z for PWMH output on channel x**

0: When fault occurs, PWMH output of channel x is forced to value defined by the bit FPVHx in PWM Fault Protection Value Register 1.

1: When fault occurs, PWMH output of channel x is forced to high-impedance state.

- **FPZLx: Fault Protection to Hi-Z for PWML output on channel x**

0: When fault occurs, PWML output of channel x is forced to value defined by the bit FPVLx in PWM Fault Protection Value Register 1.

1: When fault occurs, PWML output of channel x is forced to high-impedance state.

53.7.34 PWM Write Protection Control Register

Name: PWM_WPCR

Address: 0xF802C0E4

Access: Write-only

31	30	29	28	27	26	25	24
WPKEY							
23	22	21	20	19	18	17	16
WPKEY							
15	14	13	12	11	10	9	8
WPKEY							
7	6	5	4	3	2	1	0
WPRG5	WPRG4	WPRG3	WPRG2	WPRG1	WPRG0	WPCMD	

See Section 53.6.7 “Register Write Protection” for the list of registers that can be write-protected.

- **WPCMD: Write Protection Command**

This command is performed only if the WPKEY corresponds to 0x50574D (“PWM” in ASCII).

Value	Name	Description
0	DISABLE_SW_PROT	Disables the software write protection of the register groups of which the bit WPRGx is at ‘1’.
1	ENABLE_SW_PROT	Enables the software write protection of the register groups of which the bit WPRGx is at ‘1’.
2	ENABLE_HW_PROT	Enables the hardware write protection of the register groups of which the bit WPRGx is at ‘1’. Only a hardware reset of the PWM controller can disable the hardware write protection. Moreover, to meet security requirements, the PIO lines associated with the PWM can not be configured through the PIO interface.

- **WPRGx: Write Protection Register Group x**

0: The WPCMD command has no effect on the register group x.

1: The WPCMD command is applied to the register group x.

- **WPKEY: Write Protection Key**

Value	Name	Description
0x50574D	PASSWD	Writing any other value in this field aborts the write operation of the WPCMD field. Always reads as 0

53.7.35 PWM Write Protection Status Register

Name: PWM_WPSR

Address: 0xF802C0E8

Access: Read-only

31	30	29	28	27	26	25	24
WPVSRC							
23	22	21	20	19	18	17	16
WPVSRC							
15	14	13	12	11	10	9	8
–	–	WPHWS5	WPHWS4	WPHWS3	WPHWS2	WPHWS1	WPHWS0
7	6	5	4	3	2	1	0
WPVS	–	WPSWS5	WPSWS4	WPSWS3	WPSWS2	WPSWS1	WPSWS0

- **WPSWSx: Write Protect SW Status**

0: The SW write protection x of the register group x is disabled.

1: The SW write protection x of the register group x is enabled.

- **WPHWSx: Write Protect HW Status**

0: The HW write protection x of the register group x is disabled.

1: The HW write protection x of the register group x is enabled.

- **WPVS: Write Protect Violation Status**

0: No write protection violation has occurred since the last read of PWM_WPSR.

1: At least one write protection violation has occurred since the last read of PWM_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

- **WPVSRC: Write Protect Violation Source**

When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

53.7.36 PWM Comparison x Value Register

Name: PWM_CMPVx

Address: 0xF802C130 [0], 0xF802C140 [1], 0xF802C150 [2], 0xF802C160 [3], 0xF802C170 [4], 0xF802C180 [5], 0xF802C190 [6], 0xF802C1A0 [7]

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	CVM
23	22	21	20	19	18	17	16
CV							
15	14	13	12	11	10	9	8
CV							
7	6	5	4	3	2	1	0
CV							

Only the first 16 bits (channel counter size) of field CV are significant.

- **CV: Comparison x Value**

Define the comparison x value to be compared with the counter of the channel 0.

- **CVM: Comparison x Value Mode**

0: The comparison x between the counter of the channel 0 and the comparison x value is performed when this counter is incrementing.

1: The comparison x between the counter of the channel 0 and the comparison x value is performed when this counter is decrementing.

Note: This bit is not relevant if the counter of the channel 0 is left-aligned (CALG = 0 in PWM Channel Mode Register)

53.7.37 PWM Comparison x Value Update Register

Name: PWM_CMPVUPDx

Address: 0xF802C134 [0], 0xF802C144 [1], 0xF802C154 [2], 0xF802C164 [3], 0xF802C174 [4], 0xF802C184 [5], 0xF802C194 [6], 0xF802C1A4 [7]

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	CVMUPD

23	22	21	20	19	18	17	16
CVUPD							

15	14	13	12	11	10	9	8
CVUPD							

7	6	5	4	3	2	1	0
CVUPD							

This register acts as a double buffer for the CV and CVM values. This prevents an unexpected comparison x match.

Only the first 16 bits (channel counter size) of field CVUPD are significant.

- **CVUPD: Comparison x Value Update**

Define the comparison x value to be compared with the counter of the channel 0.

- **CVMUPD: Comparison x Value Mode Update**

0: The comparison x between the counter of the channel 0 and the comparison x value is performed when this counter is incrementing.

1: The comparison x between the counter of the channel 0 and the comparison x value is performed when this counter is decrementing.

Note: This bit is not relevant if the counter of the channel 0 is left-aligned (CALG = 0 in PWM Channel Mode Register)

CAUTION: The write of the register PWM_CMPVUPDx must be followed by a write of the register PWM_CMPMUPDx.

53.7.38 PWM Comparison x Mode Register

Name: PWM_CMPMx

Address: 0xF802C138 [0], 0xF802C148 [1], 0xF802C158 [2], 0xF802C168 [3], 0xF802C178 [4], 0xF802C188 [5], 0xF802C198 [6], 0xF802C1A8 [7]

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
CUPRCNT				CUPR			

15	14	13	12	11	10	9	8
CPRCNT				CPR			

7	6	5	4	3	2	1	0
CTR				–	–	–	CEN

- **CEN: Comparison x Enable**

0: The comparison x is disabled and can not match.

1: The comparison x is enabled and can match.

- **CTR: Comparison x Trigger**

The comparison x is performed when the value of the comparison x period counter (CPRCNT) reaches the value defined by CTR.

- **CPR: Comparison x Period**

CPR defines the maximum value of the comparison x period counter (CPRCNT). The comparison x value is performed periodically once every CPR+1 periods of the channel 0 counter.

- **CPRCNT: Comparison x Period Counter**

Reports the value of the comparison x period counter.

Note: The field CPRCNT is read-only

- **CUPR: Comparison x Update Period**

Defines the time between each update of the comparison x mode and the comparison x value. This time is equal to CUPR+1 periods of the channel 0 counter.

- **CUPRCNT: Comparison x Update Period Counter**

Reports the value of the comparison x update period counter.

Note: The field CUPRCNT is read-only

53.7.39 PWM Comparison x Mode Update Register

Name: PWM_CMPMUPDx

Address: 0xF802C13C [0], 0xF802C14C [1], 0xF802C15C [2], 0xF802C16C [3], 0xF802C17C [4], 0xF802C18C [5], 0xF802C19C [6], 0xF802C1AC [7]

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	CUPRUPD			
15	14	13	12	11	10	9	8
–	–	–	–	CPRUPD			
7	6	5	4	3	2	1	0
CTRUPD				–	–	–	CENUPD

This register acts as a double buffer for the CEN, CTR, CPR and CUPR values. This prevents an unexpected comparison x match.

- **CENUPD: Comparison x Enable Update**

0: The comparison x is disabled and can not match.

1: The comparison x is enabled and can match.

- **CTRUPD: Comparison x Trigger Update**

The comparison x is performed when the value of the comparison x period counter (CPRCNT) reaches the value defined by CTR.

- **CPRUPD: Comparison x Period Update**

CPR defines the maximum value of the comparison x period counter (CPRCNT). The comparison x value is performed periodically once every CPR+1 periods of the channel 0 counter.

- **CUPRUPD: Comparison x Update Period Update**

Defines the time between each update of the comparison x mode and the comparison x value. This time is equal to CUPR+1 periods of the channel 0 counter.

53.7.40 PWM Channel Mode Register

Name: PWM_CMRx [x=0..3]

Address: 0xF802C200 [0], 0xF802C220 [1], 0xF802C240 [2], 0xF802C260 [3]

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	PPM	DTLI	DTHI	DTE
15	14	13	12	11	10	9	8
–	–	TCTS	DPOLI	UPDS	CES	CPOL	CALG
7	6	5	4	3	2	1	0
–	–	–	–	CPRE			

This register can only be written if bits WPSWS2 and WPHWS2 are cleared in the PWM Write Protection Status Register.

- **CPRE: Channel Prescaler**

Value	Name	Description
0	MCK	Peripheral clock
1	MCK_DIV_2	Peripheral clock/2
2	MCK_DIV_4	Peripheral clock/4
3	MCK_DIV_8	Peripheral clock/8
4	MCK_DIV_16	Peripheral clock/16
5	MCK_DIV_32	Peripheral clock/32
6	MCK_DIV_64	Peripheral clock/64
7	MCK_DIV_128	Peripheral clock/128
8	MCK_DIV_256	Peripheral clock/256
9	MCK_DIV_512	Peripheral clock/512
10	MCK_DIV_1024	Peripheral clock/1024
11	CLKA	Clock A
12	CLKB	Clock B

- **CALG: Channel Alignment**

0: The period is left-aligned.

1: The period is center-aligned.

- **CPOL: Channel Polarity**

0: The OCx output waveform (output from the comparator) starts at a low level.

1: The OCx output waveform (output from the comparator) starts at a high level.

- **CES: Counter Event Selection**

The bit CES defines when the channel counter event occurs when the period is center-aligned (flag CHIDx in PWM Interrupt Status Register 1).

CALG = 0 (Left Alignment):

0/1: The channel counter event occurs at the end of the PWM period.

CALG = 1 (Center Alignment):

0: The channel counter event occurs at the end of the PWM period.

1: The channel counter event occurs at the end of the PWM period and at half the PWM period.

- **UPDS: Update Selection**

When the period is center aligned, the bit UPDS defines when the update of the duty cycle, the polarity value/mode occurs after writing the corresponding update registers.

CALG = 0 (Left Alignment):

0/1: The update always occurs at the end of the PWM period after writing the update register(s).

CALG = 1 (Center Alignment):

0: The update occurs at the next end of the PWM period after writing the update register(s).

1: The update occurs at the next end of the PWM half period after writing the update register(s).

- **DPOLI: Disabled Polarity Inverted**

0: When the PWM channel x is disabled (CHIDx(PWM_SR) = 0), the OCx output waveform is the same as the one defined by the CPOL bit.

1: When the PWM channel x is disabled (CHIDx(PWM_SR) = 0), the OCx output waveform is inverted compared to the one defined by the CPOL bit.

- **TCTS: Timer Counter Trigger Selection**

0: The comparator of the channel x (OCx) is used as the trigger source for the Timer Counter (TC).

1: The counter events of the channel x is used as the trigger source for the Timer Counter (TC).

- **DTE: Dead-Time Generator Enable**

0: The dead-time generator is disabled.

1: The dead-time generator is enabled.

- **DTHI: Dead-Time PWMHx Output Inverted**

0: The dead-time PWMHx output is not inverted.

1: The dead-time PWMHx output is inverted.

- **DTLI: Dead-Time PWMLx Output Inverted**

0: The dead-time PWMLx output is not inverted.

1: The dead-time PWMLx output is inverted.

- **PPM: Push-Pull Mode**

0: The Push-Pull mode is disabled for channel x.

1: The Push-Pull mode is enabled for channel x.

53.7.41 PWM Channel Duty Cycle Register

Name: PWM_CDTYx [x=0..3]

Address: 0xF802C204 [0], 0xF802C224 [1], 0xF802C244 [2], 0xF802C264 [3]

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
CDTY							

15	14	13	12	11	10	9	8
CDTY							

7	6	5	4	3	2	1	0
CDTY							

Only the first 16 bits (channel counter size) are significant.

- **CDTY: Channel Duty-Cycle**

Defines the waveform duty-cycle. This value must be defined between 0 and CPRD (PWM_CPRDx).

53.7.42 PWM Channel Duty Cycle Update Register

Name: PWM_CDTYUPDx [x=0..3]

Address: 0xF802C208 [0], 0xF802C228 [1], 0xF802C248 [2], 0xF802C268 [3]

Access: Write-only.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
CDTYUPD							

15	14	13	12	11	10	9	8
CDTYUPD							

7	6	5	4	3	2	1	0
CDTYUPD							

This register acts as a double buffer for the CDTY value. This prevents an unexpected waveform when modifying the waveform duty-cycle.

Only the first 16 bits (channel counter size) are significant.

- **CDTYUPD: Channel Duty-Cycle Update**

Defines the waveform duty-cycle. This value must be defined between 0 and CPRD (PWM_CPRDx).

53.7.43 PWM Channel Period Register

Name: PWM_CPRDx [x=0..3]

Address: 0xF802C20C [0], 0xF802C22C [1], 0xF802C24C [2], 0xF802C26C [3]

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
CPRD							
15	14	13	12	11	10	9	8
CPRD							
7	6	5	4	3	2	1	0
CPRD							

This register can only be written if bits WPSWS3 and WPHWS3 are cleared in the PWM Write Protection Status Register.

Only the first 16 bits (channel counter size) are significant.

- **CPRD: Channel Period**

If the waveform is left-aligned, then the output waveform period depends on the channel counter source clock and can be calculated:

– By using the PWM peripheral clock divided by a given prescaler value “X” (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

$$\frac{(X \times CPRD)}{f_{\text{peripheral clock}}}$$

– By using the PWM peripheral clock divided by a given prescaler value “X” (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

$$\frac{(X \times CPRD \times DIVA)}{f_{\text{peripheral clock}}} \text{ or } \frac{(X \times CPRD \times DIVB)}{f_{\text{peripheral clock}}}$$

If the waveform is center-aligned, then the output waveform period depends on the channel counter source clock and can be calculated:

– By using the PWM peripheral clock divided by a given prescaler value “X” (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

$$\frac{(2 \times X \times CPRD)}{f_{\text{peripheral clock}}}$$

– By using the PWM peripheral clock divided by a given prescaler value “X” (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

$$\frac{(2 \times X \times CPRD \times DIVA)}{f_{\text{peripheral clock}}} \text{ or } \frac{(2 \times X \times CPRD \times DIVB)}{f_{\text{peripheral clock}}}$$

53.7.44 PWM Channel Period Update Register

Name: PWM_CPRDUPDx [x=0..3]

Address: 0xF802C210 [0], 0xF802C230 [1], 0xF802C250 [2], 0xF802C270 [3]

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
CPRDUPD							
15	14	13	12	11	10	9	8
CPRDUPD							
7	6	5	4	3	2	1	0
CPRDUPD							

This register can only be written if bits WPSWS3 and WPHWS3 are cleared in the PWM Write Protection Status Register.

This register acts as a double buffer for the CPRD value. This prevents an unexpected waveform when modifying the waveform period.

Only the first 16 bits (channel counter size) are significant.

- **CPRDUPD: Channel Period Update**

If the waveform is left-aligned, then the output waveform period depends on the channel counter source clock and can be calculated:

– By using the PWM peripheral clock divided by a given prescaler value “X” (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

$$\frac{(X \times CPRDUPD)}{f_{\text{peripheral clock}}}$$

– By using the PWM peripheral clock divided by a given prescaler value “X” (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

$$\frac{(X \times CRPDUPD \times DIVA)}{f_{\text{peripheral clock}}} \text{ or } \frac{(X \times CRPDUPD \times DIVB)}{f_{\text{peripheral clock}}}$$

If the waveform is center-aligned, then the output waveform period depends on the channel counter source clock and can be calculated:

– By using the PWM peripheral clock divided by a given prescaler value “X” (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

$$\frac{(2 \times X \times CPRDUPD)}{f_{\text{peripheral clock}}}$$

– By using the PWM peripheral clock divided by a given prescaler value “X” (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

$$\frac{(2 \times X \times CPRDUPD \times DIVA)}{f_{\text{peripheral clock}}} \text{ or } \frac{(2 \times X \times CPRDUPD \times DIVB)}{f_{\text{peripheral clock}}}$$

53.7.45 PWM Channel Counter Register

Name: PWM_CCNTx [x=0..3]

Address: 0xF802C214 [0], 0xF802C234 [1], 0xF802C254 [2], 0xF802C274 [3]

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
CNT							
15	14	13	12	11	10	9	8
CNT							
7	6	5	4	3	2	1	0
CNT							

Only the first 16 bits (channel counter size) are significant.

- **CNT: Channel Counter Register**

Channel counter value. This register is reset when:

- the channel is enabled (writing CHIDx in the PWM_ENA register).
- the channel counter reaches CPRD value defined in the PWM_CPRDx register if the waveform is left-aligned.

53.7.46 PWM Channel Dead Time Register

Name: PWM_DTx [x=0..3]

Address: 0xF802C218 [0], 0xF802C238 [1], 0xF802C258 [2], 0xF802C278 [3]

Access: Read/Write

31	30	29	28	27	26	25	24
DTL							

23	22	21	20	19	18	17	16
DTL							

15	14	13	12	11	10	9	8
DTH							

7	6	5	4	3	2	1	0
DTH							

This register can only be written if bits WPSWS4 and WPHWS4 are cleared in the PWM Write Protection Status Register.

Only the first 16 bits (dead-time counter size) of fields DTH and DTL are significant.

- **DTH: Dead-Time Value for PWMHx Output**

Defines the dead-time value for PWMHx output. This value must be defined between 0 and the value (CPRD – CDTY) (PWM_CPRDx and PWM_CDTYx).

- **DTL: Dead-Time Value for PWMLx Output**

Defines the dead-time value for PWMLx output. This value must be defined between 0 and CDTY (PWM_CDTYx).

53.7.47 PWM Channel Dead Time Update Register

Name: PWM_DTUPDx [x=0..3]

Address: 0xF802C21C [0], 0xF802C23C [1], 0xF802C25C [2], 0xF802C27C [3]

Access: Write-only

31	30	29	28	27	26	25	24
DTLUPD							
23	22	21	20	19	18	17	16
DTLUPD							
15	14	13	12	11	10	9	8
DTHUPD							
7	6	5	4	3	2	1	0
DTHUPD							

This register can only be written if bits WPSWS4 and WPHWS4 are cleared in the PWM Write Protection Status Register.

This register acts as a double buffer for the DTH and DTL values. This prevents an unexpected waveform when modifying the dead-time values.

Only the first 16 bits (dead-time counter size) of fields DTHUPD and DTLUPD are significant.

- **DTHUPD: Dead-Time Value Update for PWMHx Output**

Defines the dead-time value for PWMHx output. This value must be defined between 0 and the value (CPRD – CDTY) (PWM_CPRDx and PWM_CDTYx). This value is applied only at the beginning of the next channel x PWM period.

- **DTLUPD: Dead-Time Value Update for PWMLx Output**

Defines the dead-time value for PWMLx output. This value must be defined between 0 and CDTY (PWM_CDTYx). This value is applied only at the beginning of the next channel x PWM period.

53.7.48 PWM Channel Mode Update Register

Name: PWM_CMUPDx [x=0..3]

Address: 0xF802C400 [0], 0xF802C420 [1], 0xF802C440 [2], 0xF802C460 [3]

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	CPOLINVUP	–	–	–	CPOLUP	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

This register can only be written if bits WPSWS2 and WPHWS2 are cleared in the PWM Write Protection Status Register.

This register acts as a double buffer for the CPOL value. This prevents an unexpected waveform when modifying the polarity value.

- **CPOLUP: Channel Polarity Update**

The write of this bit is taken into account only if the bit CPOLINVUP is written at ‘0’ at the same time.

0: The OCx output waveform (output from the comparator) starts at a low level.

1: The OCx output waveform (output from the comparator) starts at a high level.

- **CPOLINVUP: Channel Polarity Inversion Update**

If this bit is written at ‘1’, the write of the bit CPOLUP is not taken into account.

0: No effect.

1: The OCx output waveform (output from the comparator) is inverted.

53.7.49 PWM External Trigger Register

Name: PWM_ETRGx [x=1..2]

Address: 0xF802C42C [1], 0xF802C44C [2]

Access: Read/Write

31	30	29	28	27	26	25	24
RFEN	TRGSRC	TRGFILT	TRGEDGE	–	–	TRGMODE	
23	22	21	20	19	18	17	16
MAXCNT							
15	14	13	12	11	10	9	8
MAXCNT							
7	6	5	4	3	2	1	0
MAXCNT							

- **MAXCNT: Maximum Counter value**

Maximum channel x counter value measured at the TRGINx event since the last read of the register.

At the TRGINx event, if the channel x counter value is greater than the stored MAXCNT value, then MAXCNT is updated by the channel x counter value.

- **TRGMODE: External Trigger Mode**

Value	Name	Description
0	OFF	External trigger is not enabled.
1	MODE1	External PWM Reset Mode
2	MODE2	External PWM Start Mode
3	MODE3	Cycle-by-cycle Duty Mode

- **TRGEDGE: Edge Selection**

Value	Name	Description
0	FALLING_ZERO	TRGMODE = 1: TRGINx event detection on falling edge. TRGMODE = 2, 3: TRGINx active level is 0
1	RISING_ONE	TRGMODE = 1: TRGINx event detection on rising edge. TRGMODE = 2, 3: TRGINx active level is 1

- **TRGFILT: Filtered input**

0: The external trigger input x is not filtered.

1: The external trigger input x is filtered.

- **RFEN: Recoverable Fault Enable**

0: The TRGINx signal does not generate a recoverable fault.

1: The TRGINx signal generate a recoverable fault in place of the fault x input.

- **TRGSRC: Trigger Source**

0: The TRGINx signal is driven by the PWMEXTRGx input.

1: The TRGINx signal is driven by the Analog Comparator Controller.

53.7.50 PWM Leading-Edge Blanking Register

Name: PWM_LEBRx [x=1..2]

Address: 0xF802C430 [1], 0xF802C450 [2]

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	PWMHREN	PWMHFEN	PWMLREN	PWMLFEN
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	LEBDELAY						

- **LEBDELAY: Leading-Edge Blanking Delay for TRGINx**

Leading-edge blanking duration for external trigger x input. The delay is calculated according to the following formula:

LEBDELAY = ($f_{peripheral\ clock} \times$ Delay) + 1

- **PWMLFEN: PWML Falling Edge Enable**

0: Leading-edge blanking is disabled on PWMLx output falling edge.

1: Leading-edge blanking is enabled on PWMLx output falling edge.

- **PWMLREN: PWML Rising Edge Enable**

0: Leading-edge blanking is disabled on PWMLx output rising edge.

1: Leading-edge blanking is enabled on PWMLx output rising edge.

- **PWMHFEN: PWMH Falling Edge Enable**

0: Leading-edge blanking is disabled on PWMHx output falling edge.

1: Leading-edge blanking is enabled on PWMHx output falling edge.

- **PWMHREN: PWMH Rising Edge Enable**

0: Leading-edge blanking is disabled on PWMHx output rising edge.

1: Leading-edge blanking is enabled on PWMHx output rising edge.

54. Secure Fuse Controller (SFC)

54.1 Description

The Secure Fuse Controller (SFC) interfaces the system with electrical fuses in a secure way.

The default value of a fuse is logic ‘0’ (not programmed). A programmed fuse is logic ‘1’.

An electrical fuse matrix is a type of non-volatile memory. Each fuse in the matrix can be programmed only one time. They are typically used to store calibration bits for analog cells such as oscillators, configuration settings, chip identifiers or cryptographic keys.

A specific number of fuse bits are programmed by Atmel during the production tests through the test interface. The remaining 544 fuse bits are programmed by the user and by software through the user interface.

The SFC automatically reads the fuse values on startup and stores them in 32-bit registers in order to make them accessible by the software. Only fuses set to level ‘1’ are programmed.

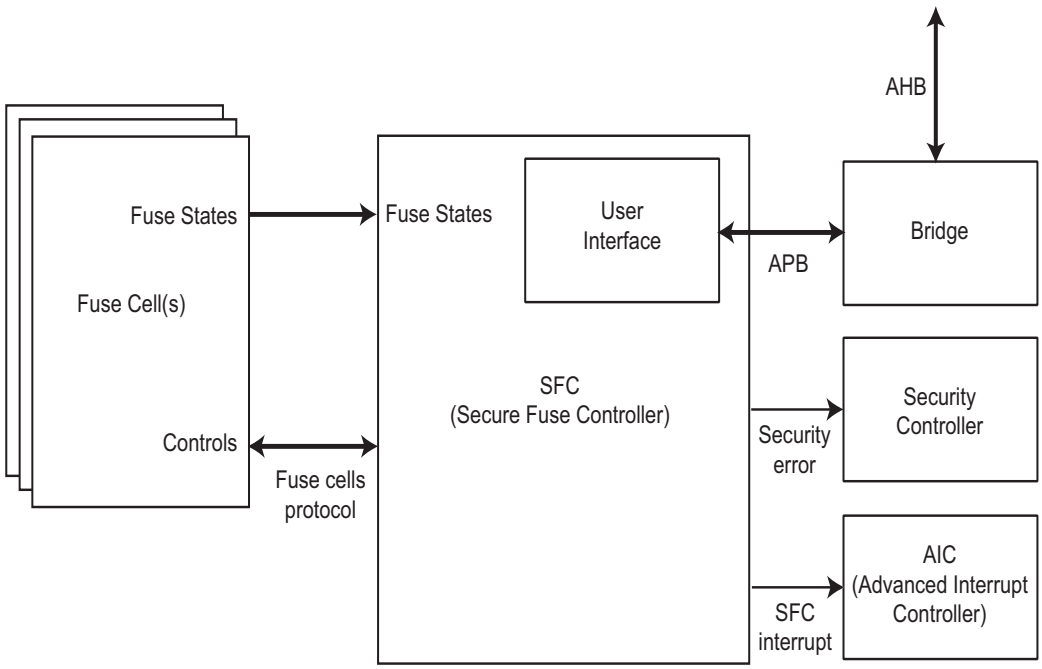
Several security mechanisms make irregular data recovery more complex to achieve.

54.2 Embedded Characteristics

- Fuse bits partitioned into two areas:
  - Atmel reserved area
  - 544-bit user area
- Program and read the fuse states by software
- Automatic check of programmed fuses
- Detection of irregular alteration of the fuse states in Atmel reserved area during startup and report
- Live detection of irregular alteration of all the fuse states and report
- Part of fuse states maskable for reading

54.3 Block Diagram

Figure 54-1. SFC Block Diagram



54.4 Functional Description

54.4.1 Accessing the SFC

Setting the write-once FUSE bit in the SFR_SECURE register disables access to the Secure Fuse Controller (SFC).

54.4.2 Fuse Partitioning

The fuses are split into a user area of 544 bits and an Atmel reserved area.

The Atmel reserved area is typically used to store calibration bits for analog cells such as oscillators, configuration settings, chip identifiers, etc. The user area fuses are programmed later on by the user.

54.4.3 Fuse Integrity Checking

The SFC automatically reads the fuses values at startup and stores them in 32-bit registers in order to make them accessible by software. At this time, the SFC checks the integrity of the fuse states in the Atmel reserved area.

If an inconsistency is detected, the CHECK error flag (named ACE for the Atmel reserved area) in the Status Register (SFC_SR) is set to ‘1’ and can trigger an interrupt. This flag is automatically cleared at ‘0’, when the Status Register (SFC_SR) is read.

54.4.4 Fuse Integrity Live Checking

The SFC automatically checks the integrity of all fuse states at every time after the startup is finished. This ensures that the fuses states cannot be changed without notice.

If an inconsistency is detected, the LCHECK error flag in the Status Register (SFC_SR) is set to ‘1’ and can trigger an interrupt. This flag is automatically cleared at ‘0’, when the Status Register (SFC_SR) is read.

54.4.5 Fuse Access

54.4.5.1 Fuse Reading

The fuse states are automatically latched at core startup and are available for reading in the Data Registers (SFC_DRx).

The fuse states of bits 0 to 31 are available in the Data Register 0 (SFC_DR0), the fuse states of bits 32 to 63 are available in the Data Register 1 (SFC_DR1) and so on.

When fuse programming is performed, the fuse states are automatically updated in the Data Registers (SFC_DRx).

54.4.5.2 Fuse Programming

All the fuses can be written by software.

The sequence of instructions to program fuses is the following:

1. Write the key code 0xFB in the Key Register (SFC_KR).
2. Write the word to program in the corresponding Data Register (SFC_DRx).
   For example, if fuses 0 to 31 must be programmed, Data Register 0 (SFC_DR0) must be written. If fuses 32 to 61 must be programmed, Data Register 1 (SFC_DR1) must be written. Only the data bits set to level ‘1’ are programmed.
3. Wait for flag PGMC to rise in the Status Register (SFC_SR) by polling or interrupt.
4. Check the value of flag PGMF: if it is set to 1, it means that the programming procedure failed. After programming, the fuses are read back in the corresponding SFC_DRx.

54.4.5.3 Fuse Masking

It is possible to mask a fuse array. Once the fuse masking is enabled, the data registers from SFC_DR20 to SFC_DR23 are read at a value of ‘0’, regardless of the fuse state (the registers that are masked depend on the SFC hardware customizing).

To activate fuse masking, the MSK bit of the SFC Mode Register (SFC_MR) must be written to level ‘1’. The MSK bit is set-only. Only a hardware reset can disable fuse masking.

The MSK bit has no effect on the programming of masked fuses.

54.4.6 Fuse Functions

The “Fuse Box Controller” section defines the fuse bits that can be used as general purpose bits when standard boot is used.

If secure boot is used, refer to the device “Secure Boot Strategy” application note included in the Secure Package.

54.5 Secure Fuse Controller (SFC) User Interface

Table 54-1. Register Mapping

Offset	Register	Name	Access	Reset
0x00	SFC Key Register	SFC_KR	Write-only	–
0x04	SFC Mode Register	SFC_MR	Read/Write	0x0
0x08–0x0C	Reserved	–	–	–
0x10	SFC Interrupt Enable Register	SFC_IER	Write-only	–
0x14	SFC Interrupt Disable Register	SFC_IDR	Write-only	–
0x18	SFC Interrupt Mask Register	SFC_IMR	Read-only	0x0
0x1C	SFC Status Register	SFC_SR	Read-only	0x0
0x20	SFC Data Register 0	SFC_DR0	Read/Write	0x0
0x24	SFC Data Register 1	SFC_DR1	Read/Write	0x0
...	...	...	...	...
0x7C	SFC Data Register 23	SFC_DR23	Read/Write	0x0
0x80–0xFC	Reserved	–	–	–

54.5.1 SFC Key Register

Name: SFC_KR

Address: 0xF804C000

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
KEY							

- **KEY: Key Code**

This field must be written with the correct key code (0xFB) prior to any write in a Data Register (SFC_DRx) in order to enable the fuse programming. For each write of SFC_DRx, this field must be written immediately before.

54.5.2 SFC Mode Register

Name: SFC_MR

Address: 0xF804C004

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	SASEL	–	–	–	MSK

- **MSK: Mask Data Registers**

0: No effect

1: The data registers from SFC_DR20 to SFC_DR23 are always read at 0x00000000.

Note: The MSK bit is set-only. Only a hardware reset can disable fuse masking.

- **SASEL: Sense Amplifier Selection**

0: Comparator type sense amplifier selected

1: Latch type sense amplifier selected

54.5.3 SFC Interrupt Enable Register

Name: SFC_IER

Address: 0xF804C010

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	ACE	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	LCHECK	–	–	PGMF	PGMC

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Enables the corresponding interrupt

- **PGMC: Programming Sequence Completed Interrupt Enable**
- **PGMF: Programming Sequence Failed Interrupt Enable**
- **LCHECK: Live Integrity Check Error Interrupt Enable**
- **ACE: Atmel Check Error Interrupt Enable**

54.5.4 SFC Interrupt Disable Register

Name: SFC_IDR

Address: 0xF804C014

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	ACE	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	LCHECK	–	–	PGMF	PGMC

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Disables the corresponding interrupt

- **PGMC: Programming Sequence Completed Interrupt Disable**
- **PGMF: Programming Sequence Failed Interrupt Disable**
- **LCHECK: Live Integrity Check Error Interrupt Disable**
- **ACE: Atmel Check Error Interrupt Disable**

54.5.5 SFC Interrupt Mask Register

Name: SFC_IMR

Address: 0xF804C018

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	ACE	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	LCHECK	–	–	PGMF	PGMC

The following configuration values are valid for all listed bit names of this register:

0: Corresponding interrupt is not enabled.

1: Corresponding interrupt is enabled.

- **PGMC: Programming Sequence Completed Interrupt Mask**
- **PGMF: Programming Sequence Failed Interrupt Mask**
- **LCHECK: Live Integrity Checking Error Interrupt Mask**
- **ACE: Atmel Check Error Interrupt Mask**

54.5.6 SFC Status Register

Name: SFC_SR

Address: 0xF804C01C

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	ACE	APLE
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	LCHECK	–	–	PGMF	PGMC

- **PGMC: Programming Sequence Completed (cleared on read)**

0: No programming sequence completion since the last read of SFC_SR.

1: At least one programming sequence completion since the last read of SFC_SR.

- **PGMF: Programming Sequence Failed (cleared on read)**

0: No programming failure occurred during last programming sequence since the last read of SFC_SR.

1: A programming failure occurred since the last read of SFC_SR.

- **LCHECK: Live Integrity Checking Error (cleared on read)**

0: No live integrity check error since the last read of SFC_SR.

1: At least one live integrity check error since the last read of SFC_SR.

- **APLE: Atmel Programming Lock Error (cleared on read)**

0: No programming attempt has been made in the Atmel locked area since the last read of SFC_SR.

1: A programming attempt has been made in the Atmel locked area since the last read of SFC_SR.

- **ACE: Atmel Check Error (cleared on read)**

0: No check error in the Atmel reserved area since the last read of SFC_SR.

1: At least one check error in the Atmel reserved area since the last read of SFC_SR.

54.5.7 SFC Data Register x

Name: SFC_DRx [x=0..23]

Address: 0xF804C020

Access: Read/Write

31	30	29	28	27	26	25	24
DATA							
23	22	21	20	19	18	17	16
DATA							
15	14	13	12	11	10	9	8
DATA							
7	6	5	4	3	2	1	0
DATA							

- **DATA: Fuse Data**

READ: Reports the state of the corresponding fuses.

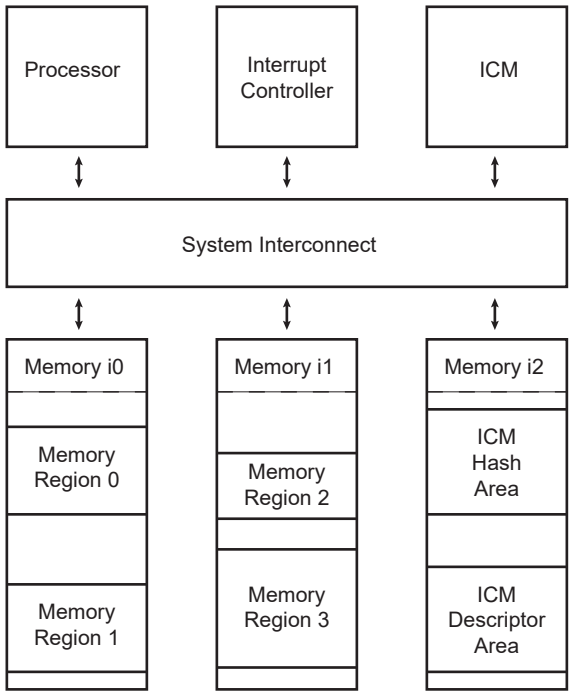
WRITE: The data to be programmed in the corresponding fuses. Only bits with a value of ‘1’ are programmed. Writing this register automatically triggers a programming sequence of the corresponding fuses. Note that a write to the Key Register (SFC_KR) with the correct key code must always precede any write to SFC_DRx.

55. Integrity Check Monitor (ICM)

55.1 Description

The Integrity Check Monitor (ICM) is a DMA controller that performs hash calculation over multiple memory regions through the use of transfer descriptors located in memory (ICM Descriptor Area). The Hash function is based on the Secure Hash Algorithm (SHA). The ICM controller integrates two modes of operation. The first one is used to hash a list of memory regions and save the digests to memory (ICM Hash Area). The second mode is an active monitoring of the memory. In that mode, the hash function is evaluated and compared to the digest located at a predefined memory address (ICM Hash Area). If a mismatch occurs, an interrupt is raised. See Figure 55-1 for an example of four-region monitoring. Hash and Descriptor areas are located in Memory instance i2, and the four regions are split in memory instances i0 and i1.

Figure 55-1. Four-region Monitoring Example



The ICM SHA engine is compliant with the American *FIPS (Federal Information Processing Standard) Publication 180-2* specification.

The following terms are concise definitions of the ICM concepts used throughout this document:

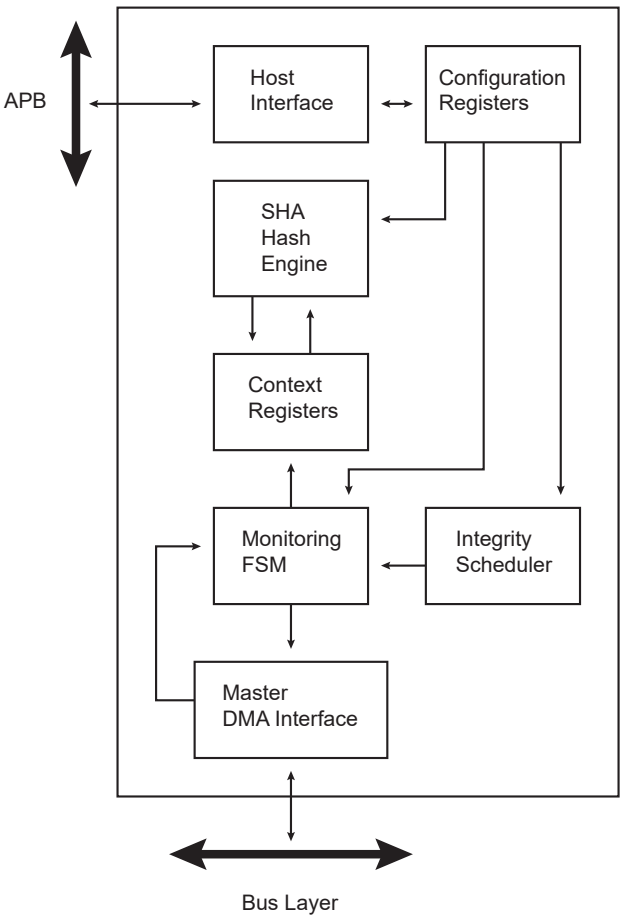
- Region—a partition of instruction or data memory space
- Region Descriptor—a data structure stored in memory, defining region attributes
- Region Attributes—region start address, region size, region SHA engine processing mode, Write Back or Compare function mode
- Context Registers—a set of ICM non-memory-mapped, internal registers which are automatically loaded, containing the attributes of the region being processed
- Main List—a list of region descriptors. Each element associates the start address of a region with a set of attributes.
- Secondary List—a linked list defined on a per region basis that describes the memory layout of the region (when the region is non-contiguous)
- Hash Area—predefined memory space where the region hash results (digest) are stored

55.2 Embedded Characteristics

- DMA AHB master interface
- Supports monitoring of up to 4 Non-Contiguous Memory Regions
- Supports block gathering through the use of linked list
- Supports Secure Hash Algorithm (SHA1, SHA224, SHA256)
- Compliant with *FIPS Publication 180-2*
- Configurable Processing Period:
  - When SHA1 algorithm is processed, the runtime period is either 85 or 209 clock cycles.
  - When SHA256 or SHA224 algorithm is processed, the runtime period is either 72 or 194 clock cycles.
- Programmable Bus burden

55.3 Block Diagram

Figure 55-2. Integrity Check Monitor Block Diagram



55.4 Product Dependencies

55.4.1 Power Management

The peripheral clock is not continuously provided to the ICM. The programmer must first enable the ICM clock in the Power Management Controller (PMC) before using the ICM.

55.4.2 Interrupt Sources

The ICM interface has an interrupt line connected to the Interrupt Controller.

Handling the ICM interrupt requires programming the interrupt controller before configuring the ICM.

Table 55-1. Peripheral IDs

Instance	ID
ICM	8

55.5 Functional Description

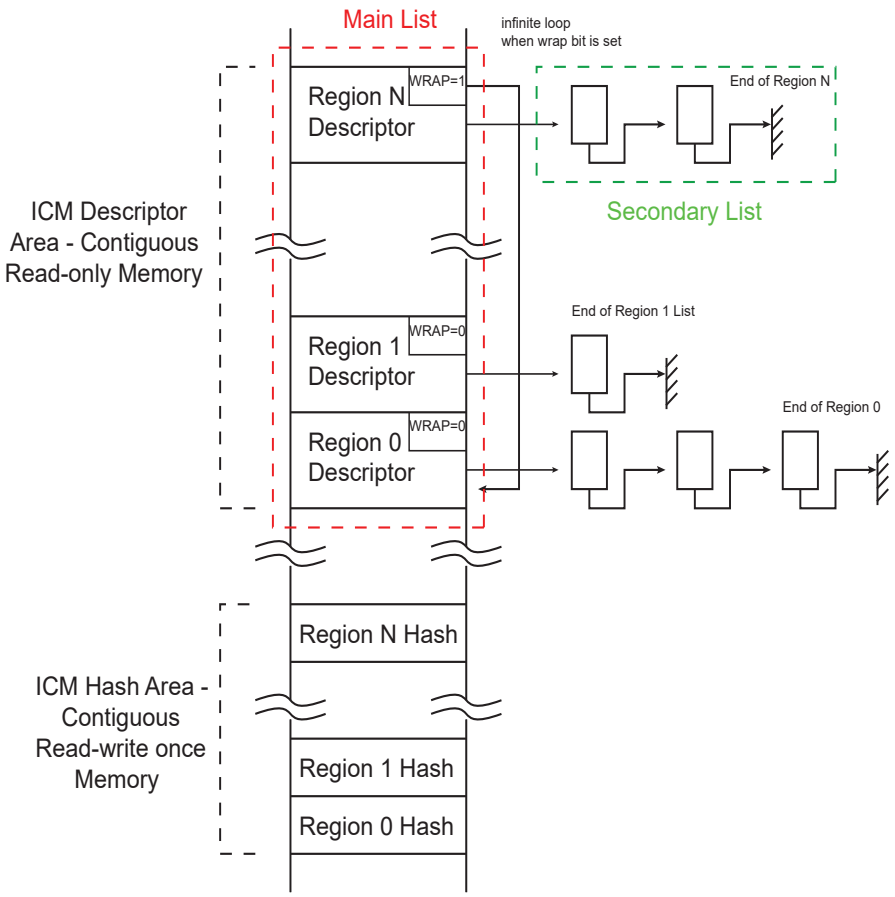
55.5.1 Overview

The Integrity Check Monitor (ICM) is a DMA controller that performs SHA-based memory hashing over memory regions. As shown in Figure 55-2, it integrates a DMA interface, a Monitoring Finite State Machine (FSM), an integrity scheduler, a set of context registers, a SHA engine, an interface for configuration and status registers.

The ICM integrates a Secure Hash Algorithm Engine (SHA). This engine requires a message padded according to FIPS180-2 specification when used as a SHA calculation unit only. Otherwise, if the ICM is used as integrated check for memory content, the padding is not mandatory. The SHA module produces an N-bit message digest each time a block is read and a processing period ends. N is 160 for SHA1, 224 for SHA224, 256 for SHA256.

When the ICM module is enabled, it sequentially retrieves a circular list of region descriptors from the memory (Main List described in Figure 55-3). Up to four regions may be monitored. Each region descriptor is composed of four words indicating the layout of the memory region (see Figure 55-4). It also contains the hashing engine configuration on a per region basis. As soon as the descriptor is loaded from the memory and context registers are updated with the data structure, the hashing operation starts. A programmable number of blocks (see TRSIZE field of the ICM_RCTRL structure member) is transferred from the memory to the SHA engine. When the desired number of blocks have been transferred, the digest is either moved to memory (Write Back function) or compared with a digest reference located in the system memory (Compare function). If a digest mismatch occurs, an interrupt is triggered if unmasked. The ICM module passes through the region descriptor list until the end of the list marked by an End of List bit set to one. To continuously monitor the list of regions, the WRAP bit must be set to one in the last data structure.

Figure 55-3. ICM Region Descriptor and Hash Areas



Each region descriptor supports gathering of data through the use of the Secondary List. Unlike the Main List, the Secondary List cannot modify the configuration attributes of the region. When the end of the Secondary List has been encountered, the ICM returns to the Main List. Memory integrity monitoring can be considered as a background service and the mandatory bandwidth shall be very limited. In order to limit the ICM memory bandwidth, use the BBC field of the ICM_CFG register to control ICM memory load.

Figure 55-4. Region Descriptor

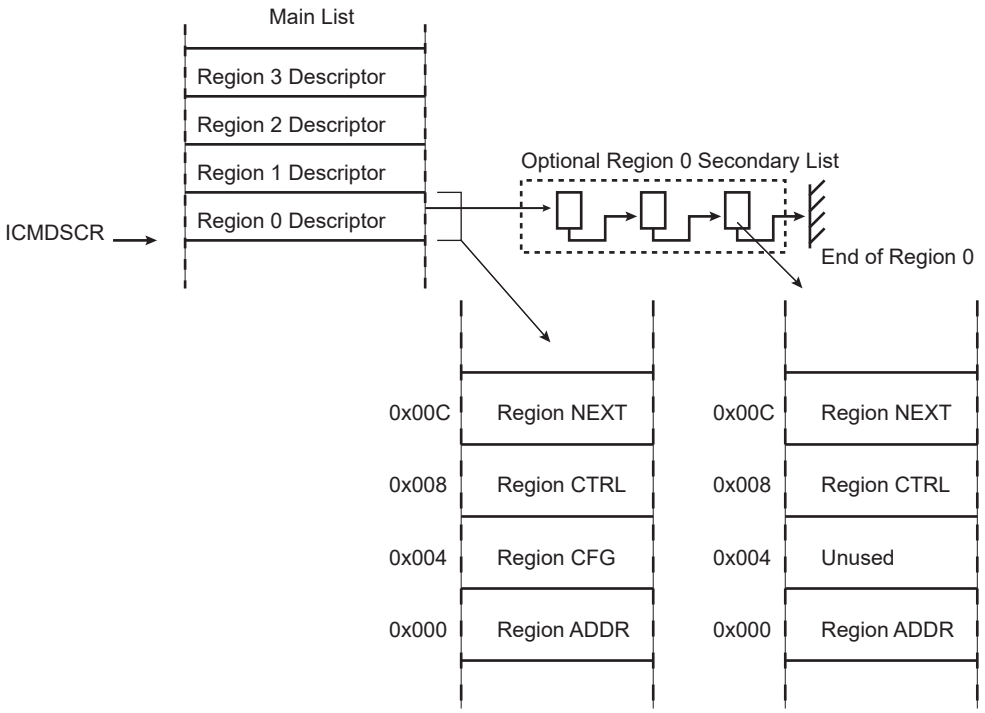
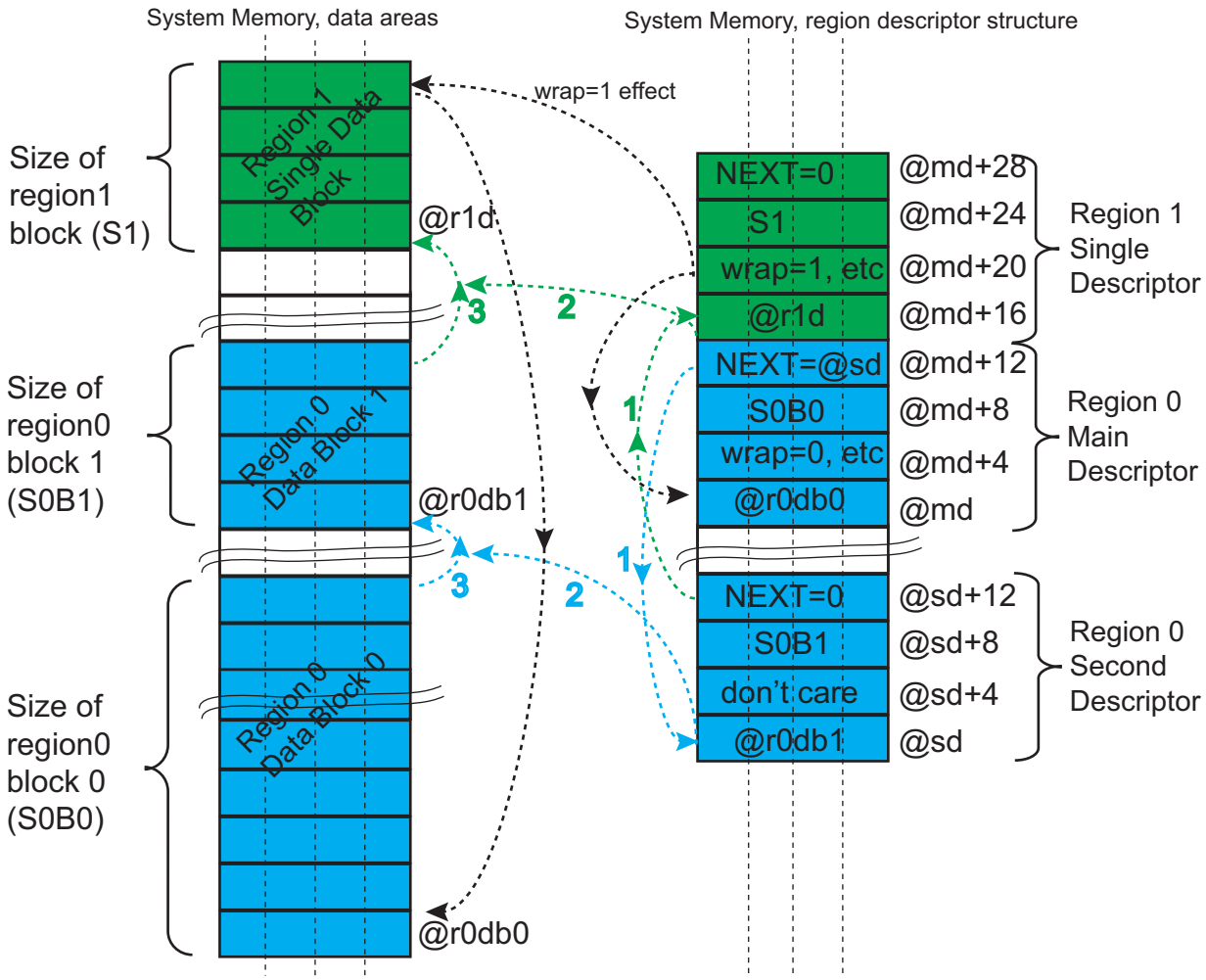


Figure 55-5 shows an example of the mandatory ICM settings to monitor three memory data blocks of the system memory (defined as two regions) with one region being not contiguous (two separate areas) and one contiguous memory area. For each said region, the SHA algorithm may be independently selected (different for each region). The wrap allows continuous monitoring.

Figure 55-5. Example: Monitoring of 3 Memory Data Blocks (Defined as 2 Regions)



55.5.2 ICM Region Descriptor Structure

The ICM Region Descriptor Area is a contiguous area of system memory that the controller and the processor can access. When the ICM controller is activated, the controller performs a descriptor fetch operation at *(ICM_DSCR) address. If the Main List contains more than one descriptor (i.e., more than one region is to be monitored), the fetch address is *(ICM_DSCR) + (RID<<4) where RID is the region identifier.

Table 55-2. Region Descriptor Structure (Main List)

Offset	Structure Member	Name
ICM_DSCR+0x000+RID*(0x10)	ICM Region Start Address	ICM_RADDR
ICM_DSCR+0x004+RID*(0x10)	ICM Region Configuration	ICM_RCFG
ICM_DSCR+0x008+RID*(0x10)	ICM Region Control	ICM_RCTRL
ICM_DSCR+0x00C+RID*(0x10)	ICM Region Next Address	ICM_RNEXT

55.5.2.1 ICM Region Start Address Structure Member

Name: ICM_RADDR

Address: ICM_DSCR+0x000+RID*(0x10)

Access: Read/Write

31	30	29	28	27	26	25	24
RADDR							

23	22	21	20	19	18	17	16
RADDR							

15	14	13	12	11	10	9	8
RADDR							

7	6	5	4	3	2	1	0
RADDR							

- **RADDR: Region Start Address**

This field indicates the first byte address of the region.

55.5.2.2 ICM Region Configuration Structure Member

Name: ICM_RCFG

Address: ICM_DSCR+0x004+RID*(0x10)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	ALGO			–	PROCDLY	SUIEN	ECIEN
7	6	5	4	3	2	1	0
WCIEN	BEIEN	DMIEN	RHIEN	–	EOM	WRAP	CDWBN

- **CDWBN: Compare Digest or Write Back Digest**

0: The digest is written to the Hash area.

1: The digest value is compared to the digest stored in the Hash area.

- **WRAP: Wrap Command**

0: The next region descriptor address loaded is the current region identifier descriptor address incremented by 0x10.

1: The next region descriptor address loaded is ICM_DSCR.

- **EOM: End Of Monitoring**

0: The current descriptor does not terminate the monitoring.

1: The current descriptor terminates the Main List. WRAP bit value has no effect.

- **RHIEN: Region Hash Completed Interrupt Disable (Default Enabled)**

0: The ICM_ISR RHC[*i*] flag is set when the field NEXT = 0 in a descriptor of the main or second list.

1: The ICM_ISR RHC[*i*] flag remains cleared even if the setting condition is met.

- **DMIEN: Digest Mismatch Interrupt Disable (Default Enabled)**

0: The ICM_ISR RBE[*i*] flag is set when the hash value just calculated from the processed region differs from expected hash value.

1: The ICM_ISR RBE[*i*] flag remains cleared even if the setting condition is met.

- **BEIEN: Bus Error Interrupt Disable (Default Enabled)**

0: The flag is set when an error is reported on the system bus by the bus MATRIX.

1: The flag remains cleared even if the setting condition is met.

- **WCIEN: Wrap Condition Interrupt Disable (Default Enabled)**

0: The ICM_ISR RWC[*i*] flag is set when the WRAP bit is set in a descriptor of the main list.

1: The ICM_ISR RWC[*i*] flag remains cleared even if the setting condition is met.

- **ECIEN: End Bit Condition Interrupt (Default Enabled)**

0: The ICM_ISR REC[*i*] flag is set when the descriptor having the EOM bit set is processed.

1: The ICM_ISR REC[*i*] flag remains cleared even if the setting condition is met.

- **SUIEN: Monitoring Status Updated Condition Interrupt (Default Enabled)**

0: The ICM_ISR RSU[*i*] flag is set when the corresponding descriptor is loaded from memory to ICM.

1: The ICM_ISR RSU[*i*] flag remains cleared even if the setting condition is met.

- **PROCDLY: Processing Delay**

Value	Name	Description
0	SHORTEST	SHA processing runtime is the shortest one
1	LONGEST	SHA processing runtime is the longest one

When SHA1 algorithm is processed, the runtime period is either 85 or 209 clock cycles.

When SHA256 or SHA224 algorithm is processed, the runtime period is either 72 or 194 clock cycles.

- **ALGO: SHA Algorithm**

Value	Name	Description
0	SHA1	SHA1 algorithm processed
1	SHA256	SHA256 algorithm processed
4	SHA224	SHA224 algorithm processed

Values which are not listed in the table must be considered as “reserved”.

55.5.2.3 ICM Region Control Structure Member

Name: ICM_RCTRL

Address: ICM_DSCR+0x008+RID*(0x10)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
TRSIZE							
7	6	5	4	3	2	1	0
TRSIZE							

- **TRSIZE: Transfer Size for the Current Chunk of Data**

55.5.2.4 ICM Region Next Address Structure Member

Name: ICM_RNEXT

Address: ICM_DSCR+0x00C+RID*(0x10)

Access: Read/Write

31	30	29	28	27	26	25	24
NEXT							
23	22	21	20	19	18	17	16
NEXT							
15	14	13	12	11	10	9	8
NEXT							
7	6	5	4	3	2	1	0
NEXT					–	–	–

- **NEXT: Region Transfer Descriptor Next Address**

When configured to 0, this field indicates that the current descriptor is the last descriptor of the Secondary List, otherwise it points at a new descriptor of the Secondary List.

55.5.3 ICM Hash Area

The ICM Hash Area is a contiguous area of system memory that the controller and the processor can access. The physical location is configured in the ICM hash area start address register. This address is a multiple of 128 bytes. If the CDWBN bit of the context register is cleared (i.e., Write Back activated), the ICM controller performs a digest write operation at the following starting location: *(ICM_HASH) + (RID<<5), where RID is the current region context identifier. If the CDWBN bit of the context register is set (i.e., Digest Comparison activated), the ICM controller performs a digest read operation at the same address.

55.5.3.1 Message Digest Example

Considering the following **512 bits message** (example given in FIPS 180-2):

"61626380000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000018"

The message is written to memory in a Little Endian (LE) system architecture.

Table 55-3. 512 bits Message Memory Mapping

Memory Address	Address Offset / Byte Lane							
	0x7 / 63:56	0x6 / 55:48	0x5 / 47:40	0x4 / 39:32	0x3 / 31:24	0x2 / 23:16	0x1 / 15:8	0x0 / 7:0
0x000	00	00	00	00	80	63	62	61
0x008–0x030	00	00	00	00	00	00	00	00
0x038	18	00	00	00	00	00	00	00

The digest is stored at the memory location pointed at by the ICM_HASH pointer with a Region Offset.

Table 55-4. LE Resulting SHA-160 Message Digest Memory Mapping

Memory Address	Address Offset / Byte Lane							
	0x7 / 63:56	0x6 / 55:48	0x5 / 47:40	0x4 / 39:32	0x3 / 31:24	0x2 / 23:16	0x1 / 15:8	0x0 / 7:0
0x000	6a	81	06	47	36	3e	99	a9
0x008	6c	c2	50	78	71	25	3e	ba
0x010	–	–	–	–	9d	d8	d0	9c

Table 55-5. Resulting SHA-256 Message Digest Memory Mapping

Memory Address	Address Offset / Byte Lane							
	0x7 / 63:56	0x6 / 55:48	0x5 / 47:40	0x4 / 39:32	0x3 / 31:24	0x2 / 23:16	0x1 / 15:8	0x0 / 7:0
0x000	ea	cf	01	8f	bf	16	78	ba
0x008	23	22	ae	5d	de	40	41	41
0x010	9c	7a	17	96	a3	61	03	b0
0x018	ad	15	00	f2	61	ff	10	b4

Considering the following **1024 bits message** (example given in FIPS 180-2):

"6162638000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000018"

The message is written to memory in a Little Endian (LE) system architecture.

Table 55-6. 1024 bits Message Memory Mapping

Memory Address	Address Offset / Byte Lane							
	0x7 / 63:56	0x6 / 55:48	0x5 / 47:40	0x4 / 39:32	0x3 / 31:24	0x2 / 23:16	0x1 / 15:8	0x0 / 7:0
0x000	00	00	00	00	80	63	62	61
0x008–0x070	00	00	00	00	00	00	00	00
0x078	18	00	00	00	00	00	00	00

55.5.4 Using ICM as SHA Engine

The ICM can be configured to only calculate a SHA1, SHA224, SHA256 digest value.

55.5.4.1 Settings for Simple SHA Calculation

The start address of the system memory containing the data to hash must be configured in the transfer descriptor of the DMA embedded in the ICM.

The transfer descriptor is a system memory area integer multiple of 4 x 32-bit word and the start address of the descriptor must be configured in ICM_DSCR (the start address must be aligned on 64-bytes; six LSB must be cleared). If the data to hash is already padded according to SHA standards, only a single descriptor is required, and the EOM bit of ICM_RCFG must be written to 1. If the data to hash does not contain a padding area, it is possible to define the padding area in another system memory location, the ICM can be configured to automatically jump from a memory area to another one by configuring the descriptor register ICM_RNEXT with a value that differs from 0. Configuring the field NEXT of the ICM_RNEXT with the start address of the padding area forces the ICM to concatenate both areas, thus providing the SHA result from the start address of the hash area configured in ICM_HASH.

Whether the system memory is configured as a single or multiple data block area, the bits CDWBN and WRAP must be cleared in the region descriptor structure member ICM_RCFG. The bits WBDIS, EOMDIS, SLBDIS must be cleared in ICM_CFG.

The bits RHIEN or ECIEN must be written to 1 in the region descriptor structure member ICM_RCTRL. The flag RHC[*i*], *i* being the region index, is set (if RHIEN is set) when the hash result is available at address defined in ICM_HASH. The flag REC[*i*], *i* being the region index, is set (if ECIEN is set) when the hash result is available at the address defined in ICM_HASH.

An interrupt is generated if the bit RHC[*i*] is written to 1 in the ICM_IER (if RHC[*i*] is set in ICM_RCTRL of region *i*) or if the bit REC[*i*] is written to 1 in the ICM_IER (if REC[*i*] is set in ICM_RCTRL of region *i*).

55.5.4.2 Processing Period

The SHA engine processing period can be configured.

The short processing period allows to allocate bandwidth to the SHA module whereas the long processing period allocates more bandwidth on the system bus to other applications.

In SHA mode, the shortest processing period is 85 clock cycles + 2 clock cycles for start command synchronization. The longest period is 209 clock cycles + 2 clock cycles.

In SHA256 and SHA224 modes, the shortest processing period is 72 clock cycles + 2 clock cycles for start command synchronization. The longest period is 194 clock cycles + 2 clock cycles.

55.5.5 ICM Automatic Monitoring Mode

The ASCD bit of the ICM_CFG register is used to activate the ICM Automatic mode. When ICM_CFG.ASCD is set, the ICM performs the following actions:

- The ICM controller passes through the Main List once with CDWBN bit in the context register at 0 (i.e., Write Back activated) and EOM bit in context register at 0.
- When WRAP = 1 in ICM_RCFG, the ICM controller enters active monitoring with CDWBN bit in context register now set and EOM bit in context register cleared. Bits CDWBN and EOM in ICM_RCFG have no effect.

55.5.6 Programming the ICM for Multiple Regions

Table 55-7. Region Attributes

Transfer Type		Main List	ICM_RCFG			ICM_RNEXT	Comments
			CDWBN	WRAP	EOM	NEXT	
Single Region	Contiguous list of blocks Digest written to memory Monitoring disabled	1 item	0	0	1	0	The Main List contains only one descriptor. The Secondary List is empty for that descriptor. The digest is computed and saved to memory.
	Non-contiguous list of blocks Digest written to memory Monitoring disabled	1 item	0	0	1	Secondary List address of the current region identifier	The Main List contains only one descriptor. The Secondary List describes the layout of the non-contiguous region.
	Contiguous list of blocks Digest comparison enabled Monitoring enabled	1 item	1	1	0	0	When the hash computation is terminated, the digest is compared with the one saved in memory.
Multiple Regions	Contiguous list of blocks Digest written to memory Monitoring disabled	More than one item	0	0	1 for the last, 0 otherwise	0	ICM passes through the list once.
	Contiguous list of blocks Digest comparison is enabled Monitoring is enabled	More than one item	1	1 for the last, 0 otherwise	0	0	ICM performs active monitoring of the regions. If a mismatch occurs, an interrupt is raised.
	Non-contiguous list of blocks Digest is written to memory Monitoring is disabled	More than one item	0	0	1	Secondary List address	ICM performs hashing and saves digests to the Hash area.
	Non-contiguous list of blocks Digest comparison is enabled Monitoring is enabled	More than one item	1	1	0	Secondary List address	ICM performs data gathering on a per region basis.

55.5.7 Security Features

When an undefined register access occurs, the URAD bit in the Interrupt Status Register (ICM_ISR) is set if unmasked. Its source is then reported in the Undefined Access Status Register (ICM_UASR). Only the first undefined register access is available through the ICM_UASR.URAT field.

Several kinds of unspecified register accesses can occur:

- Unspecified structure member set to one detected when the descriptor is loaded
- Configuration register (ICM_CFG) modified during active monitoring
- Descriptor register (ICM_DSCR) modified during active monitoring
- Hash register (ICM_HASH) modified during active monitoring
- Write-only register read access

The URAD bit and the URAT field can only be reset by writing a 1 to the ICM_CTRL.SWRST bit.

55.6 Integrity Check Monitor (ICM) User Interface

Table 55-8. Register Mapping

Offset	Register	Name	Access	Reset
0x00	Configuration Register	ICM_CFG	Read/Write	0x0
0x04	Control Register	ICM_CTRL	Write-only	–
0x08	Status Register	ICM_SR	Read-only	–
0x0C	Reserved	–	–	–
0x10	Interrupt Enable Register	ICM_IER	Write-only	–
0x14	Interrupt Disable Register	ICM_IDR	Write-only	–
0x18	Interrupt Mask Register	ICM_IMR	Read-only	0x0
0x1C	Interrupt Status Register	ICM_ISR	Read-only	0x0
0x20	Undefined Access Status Register	ICM_UASR	Read-only	0x0
0x24–0x2C	Reserved	–	–	–
0x30	Region Descriptor Area Start Address Register	ICM_DSCR	Read/Write	0x0
0x34	Region Hash Area Start Address Register	ICM_HASH	Read/Write	0x0
0x38	User Initial Hash Value 0 Register	ICM_UIHVAL0	Write-only	–
...	...	...	...	...
0x54	User Initial Hash Value 7	ICM_UIHVAL7	Write-only	–
0x78–0xE8	Reserved	–	–	–
0xEC–0xFC	Reserved	–	–	–

55.6.1 ICM Configuration Register

Name: ICM_CFG

Address: 0xF8040000

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
UALGO			UIHASH	–	–	DUALBUFF	ASCD
7	6	5	4	3	2	1	0
BBC				–	SLBDIS	EOMDIS	WBDIS

- **WBDIS: Write Back Disable**

0: Write Back Operations are permitted.

1: Write Back Operations are forbidden. Context register CDWBN bit is internally set to one and cannot be modified by a linked list element. The CDWBN bit of the ICM_RCFG structure member has no effect.

When ASCD bit of the ICM_CFG register is set, WBDIS bit value has no effect.

- **EOMDIS: End of Monitoring Disable**

0: End of Monitoring is permitted

1: End of Monitoring is forbidden. The EOM bit of the ICM_RCFG structure member has no effect.

- **SLBDIS: Secondary List Branching Disable**

0: Branching to the Secondary List is permitted.

1: Branching to the Secondary List is forbidden. The NEXT field of the ICM_RNEXT structure member has no effect and is always considered as zero.

- **BBC: Bus Burden Control**

This field is used to control the burden of the ICM system bus. The number of system clock cycles between the end of the current processing and the next block transfer is set to 2^{BBC}. Up to 32,768 cycles can be inserted.

- **ASCD: Automatic Switch To Compare Digest**

0: Automatic mode is disabled.

1: When this mode is enabled, the ICM controller automatically switches to active monitoring after the first Main List pass. Both CDWBN and WBDIS bits have no effect. A one must be written to the EOM bit in ICM_RCFG to terminate the monitoring.

- **DUALBUFF: Dual Input Buffer**

0: Dual Input Buffer mode is disabled.

1: Dual Input Buffer mode is enabled (better performances, higher bandwidth required on system bus).

- **UIHASH: User Initial Hash Value**

0: The secure hash standard provides the initial hash value.

1: The initial hash value is programmable. Field UALGO provides the SHA algorithm. The ALGO field of the ICM_RCFG structure member has no effect.

- **UALGO: User SHA Algorithm**

Value	Name	Description
0	SHA1	SHA1 algorithm processed
1	SHA256	SHA256 algorithm processed
4	SHA224	SHA224 algorithm processed

55.6.2 ICM Control Register

Name: ICM_CTRL

Address: 0xF8040004

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
RMEN				RMDIS			
7	6	5	4	3	2	1	0
REHASH				–	SWRST	DISABLE	ENABLE

- **ENABLE: ICM Enable**

0: No effect

1: When set to one, the ICM controller is activated.

- **DISABLE: ICM Disable Register**

0: No effect

1: The ICM controller is disabled. If a region is active, this region is terminated.

- **SWRST: Software Reset**

0: No effect

1: Resets the ICM controller.

- **REHASH: Recompute Internal Hash**

0: No effect

1: When REHASH[*i*] is set to one, Region *i* digest is re-computed. This bit is only available when region monitoring is disabled.

- **RMDIS: Region Monitoring Disable**

0: No effect

1: When bit RMDIS[*i*] is set to one, the monitoring of region with identifier *i* is disabled.

- **RMEN: Region Monitoring Enable**

0: No effect

1: When bit RMEN[*i*] is set to one, the monitoring of region with identifier *i* is activated.

Monitoring is activated by default.

55.6.3 ICM Status Register

Name: ICM_SR

Address: 0xF8040008

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
RMDIS				RAWRMDIS			
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	ENABLE

- **ENABLE: ICM Controller Enable Register**

0: ICM controller is disabled

1: ICM controller is activated

- **RAWRMDIS: Region Monitoring Disabled Raw Status**

0: Region *i* monitoring has been activated by writing a 1 in RMEN[*i*] of ICM_CTRL

1: Region *i* monitoring has been deactivated by writing a 1 in RMDIS[*i*] of ICM_CTRL

- **RMDIS: Region Monitoring Disabled Status**

0: Region *i* is being monitored (occurs after integrity check value has been calculated and written to Hash area)

1: Region *i* monitoring is not being monitored

55.6.4 ICM Interrupt Enable Register

Name: ICM_IER

Address: 0xF8040010

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	URAD
23	22	21	20	19	18	17	16
RSU				REC			
15	14	13	12	11	10	9	8
RWC				RBE			
7	6	5	4	3	2	1	0
RDM				RHC			

- **RHC: Region Hash Completed Interrupt Enable**

0: No effect

1: When RHC[*i*] is set to one, the Region *i* Hash Completed interrupt is enabled.

- **RDM: Region Digest Mismatch Interrupt Enable**

0: No effect

1: When RDM[*i*] is set to one, the Region *i* Digest Mismatch interrupt is enabled.

- **RBE: Region Bus Error Interrupt Enable**

0: No effect

1: When RBE[*i*] is set to one, the Region *i* Bus Error interrupt is enabled.

- **RWC: Region Wrap Condition detected Interrupt Enable**

0: No effect

1: When RWC[*i*] is set to one, the Region *i* Wrap Condition interrupt is enabled.

- **REC: Region End bit Condition Detected Interrupt Enable**

0: No effect

1: When REC[*i*] is set to one, the region *i* End bit Condition interrupt is enabled.

- **RSU: Region Status Updated Interrupt Disable**

0: No effect

1: When RSU[*i*] is set to one, the region *i* Status Updated interrupt is enabled.

- **URAD: Undefined Register Access Detection Interrupt Enable**

0: No effect

1: The Undefined Register Access interrupt is enabled.

55.6.5 ICM Interrupt Disable Register

Name: ICM_IDR

Address: 0xF8040014

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	URAD

23	22	21	20	19	18	17	16
RSU				REC			

15	14	13	12	11	10	9	8
RWC				RBE			

7	6	5	4	3	2	1	0
RDM				RHC			

- **RHC: Region Hash Completed Interrupt Disable**

0: No effect

1: When RHC[*i*] is set to one, the Region *i* Hash Completed interrupt is disabled.

- **RDM: Region Digest Mismatch Interrupt Disable**

0: No effect

1: When RDM[*i*] is set to one, the Region *i* Digest Mismatch interrupt is disabled.

- **RBE: Region Bus Error Interrupt Disable**

0: No effect

1: When RBE[*i*] is set to one, the Region *i* Bus Error interrupt is disabled.

- **RWC: Region Wrap Condition Detected Interrupt Disable**

0: No effect

1: When RWC[*i*] is set to one, the Region *i* Wrap Condition interrupt is disabled.

- **REC: Region End bit Condition detected Interrupt Disable**

0: No effect

1: When REC[*i*] is set to one, the region *i* End bit Condition interrupt is disabled.

- **RSU: Region Status Updated Interrupt Disable**

0: No effect

1: When RSU[*i*] is set to one, the region *i* Status Updated interrupt is disabled.

- **URAD: Undefined Register Access Detection Interrupt Disable**

0: No effect

1: Undefined Register Access Detection interrupt is disabled.

55.6.6 ICM Interrupt Mask Register

Name: ICM_IMR

Address: 0xF8040018

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	URAD

23	22	21	20	19	18	17	16
RSU				REC			

15	14	13	12	11	10	9	8
RWC				RBE			

7	6	5	4	3	2	1	0
RDM				RHC			

- **RHC: Region Hash Completed Interrupt Mask**

0: When RHC[*i*] is set to zero, the interrupt is disabled for region i.

1: When RHC[*i*] is set to one, the interrupt is enabled for region i.

- **RDM: Region Digest Mismatch Interrupt Mask**

0: When RDM[*i*] is set to zero, the interrupt is disabled for region i.

1: When RDM[*i*] is set to one, the interrupt is enabled for region i.

- **RBE: Region Bus Error Interrupt Mask**

0: When RBE[*i*] is set to zero, the interrupt is disabled for region i.

1: When RBE[*i*] is set to one, the interrupt is enabled for region i.

- **RWC: Region Wrap Condition Detected Interrupt Mask**

0: When RWC[*i*] is set to zero, the interrupt is disabled for region i.

1: When RWC[*i*] is set to one, the interrupt is enabled for region i.

- **REC: Region End bit Condition Detected Interrupt Mask**

0: When REC[*i*] is set to zero, the interrupt is disabled for region i.

1: When REC[*i*] is set to one, the interrupt is enabled for region i.

- **RSU: Region Status Updated Interrupt Mask**

0: When RSU[*i*] is set to zero, the interrupt is disabled for region i.

1: When RSU[*i*] is set to one, the interrupt is enabled for region i.

- **URAD: Undefined Register Access Detection Interrupt Mask**

0: Interrupt is disabled

1: Interrupt is enabled.

55.6.7 ICM Interrupt Status Register

Name: ICM_ISR

Address: 0xF804001C

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	URAD

23	22	21	20	19	18	17	16
RSU				REC			

15	14	13	12	11	10	9	8
RWC				RBE			

7	6	5	4	3	2	1	0
RDM				RHC			

- **RHC: Region Hash Completed**

When RHC[*i*] is set, it indicates that the ICM has completed the region with identifier *i*.

- **RDM: Region Digest Mismatch**

When RDM[*i*] is set, it indicates that there is a digest comparison mismatch between the hash value of the region with identifier *i* and the reference value located in the Hash Area.

- **RBE: Region Bus Error**

When RBE[*i*] is set, it indicates that a bus error has been detected while hashing memory region *i*.

- **RWC: Region Wrap Condition Detected**

When RWC[*i*] is set, it indicates that a wrap condition has been detected.

- **REC: Region End bit Condition Detected**

When REC[*i*] is set, it indicates that an end bit condition has been detected.

- **RSU: Region Status Updated Detected**

When RSU[*i*] is set, it indicates that a region status updated condition has been detected.

- **URAD: Undefined Register Access Detection Status**

0: No undefined register access has been detected since the last SWRST.

1: At least one undefined register access has been detected since the last SWRST.

The URAD bit is only reset by the SWRST bit in the ICM_CTRL register.

The URAT field in the ICM_UASR indicates the unspecified access type.

55.6.8 ICM Undefined Access Status Register

Name: ICM_UASR

Address: 0xF8040020

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	URAT		

- **URAT: Undefined Register Access Trace**

Value	Name	Description
0	UNSPEC_STRUCT_MEMBER	Unspecified structure member set to one detected when the descriptor is loaded.
1	ICM_CFG_MODIFIED	ICM_CFG modified during active monitoring.
2	ICM_DSCR_MODIFIED	ICM_DSCR modified during active monitoring.
3	ICM_HASH_MODIFIED	ICM_HASH modified during active monitoring
4	READ_ACCESS	Write-only register read access

Only the first Undefined Register Access Trace is available through the URAT field.

The URAT field is only reset by the SWRST bit in the ICM_CTRL register.

55.6.9 ICM Descriptor Area Start Address Register

Name: ICM_DSCR

Address: 0xF8040030

Access: Read/Write

31	30	29	28	27	26	25	24
DASA							

23	22	21	20	19	18	17	16
DASA							

15	14	13	12	11	10	9	8
DASA							

7	6	5	4	3	2	1	0
DASA		–	–	–	–	–	–

- **DASA: Descriptor Area Start Address**

The start address is a multiple of the total size of the data structure (64 bytes).

55.6.10 ICM Hash Area Start Address Register

Name: ICM_HASH

Address: 0xF8040034

Access: Read/Write

31	30	29	28	27	26	25	24
HASA							
23	22	21	20	19	18	17	16
HASA							
15	14	13	12	11	10	9	8
HASA							
7	6	5	4	3	2	1	0
HASA	–	–	–	–	–	–	–

- **HASA: Hash Area Start Address**

This field points at the Hash memory location. The address must be a multiple of 128 bytes.

55.6.11 ICM User Initial Hash Value Register

Name: ICM_UIHVALx [x=0..7]

Address: 0xF8040038

Access: Write-only

31	30	29	28	27	26	25	24
VAL							
23	22	21	20	19	18	17	16
VAL							
15	14	13	12	11	10	9	8
VAL							
7	6	5	4	3	2	1	0
VAL							

- **VAL: Initial Hash Value**

When UIHASH bit of IMC_CFG register is set, the Initial Hash Value is user-programmable.

To meet the desired standard, use the following example values.

For ICM_UIHVAL0 field:

Example	Comment
0x67452301	SHA1 algorithm
0xC1059ED8	SHA224 algorithm
0x6A09E667	SHA256 algorithm

For ICM_UIHVAL1 field:

Example	Comment
0xEFCDAB89	SHA1 algorithm
0x367CD507	SHA224 algorithm
0xBB67AE85	SHA256 algorithm

For ICM_UIHVAL2 field:

Example	Comment
0x98BADCFE	SHA1 algorithm
0x3070DD17	SHA224 algorithm
0x3C6EF372	SHA256 algorithm

For ICM_UIHVAL3 field:

Example	Comment
0x10325476	SHA1 algorithm
0xF70E5939	SHA224 algorithm
0xA54FF53A	SHA256 algorithm

For ICM_UIHVAL4 field:

Example	Comment
0xC3D2E1F0	SHA1 algorithm
0xFFC00B31	SHA224 algorithm
0x510E527F	SHA256 algorithm

For ICM_UIHVAL5 field:

Example	Comment
0x68581511	SHA224 algorithm
0x9B05688C	SHA256 algorithm

For ICM_UIHVAL6 field:

Example	Comment
0x64F98FA7	SHA224 algorithm
0x1F83D9AB	SHA256 algorithm

For ICM_UIHVAL7 field:

Example	Comment
0xBEFA4FA4	SHA224 algorithm
0x5BE0CD19	SHA256 algorithm

Example of Initial Value for SHA-1 Algorithm

Register Address	Address Offset / Byte Lane			
	0x3 / 31:24	0x2 / 23:16	0x1 / 15:8	0x0 / 7:0
0x000 ICM_UIHVAL0	01	23	45	67
0x004 ICM_UIHVAL1	89	ab	cd	ef
0x008 ICM_UIHVAL2	fe	dc	ba	98
0x00C ICM_UIHVAL3	76	54	32	10
0x010 ICM_UIHVAL4	f0	e1	d2	c3

56. Advanced Encryption Standard Bridge (AESB)

56.1 Description

The Advanced Encryption Standard Bridge (AESB) is intended to provide on-the-fly off-chip memory encryption/decryption compliant with the American *FIPS (Federal Information Processing Standard) Publication 197* specification.

The AESB supports three confidentiality modes of operation for symmetrical key block cipher algorithms (ECB, CBC and CTR), as specified in the *NIST Special Publication 800-38A Recommendation*.

The 128-bit key is stored in four 32-bit registers (AESB_KEYWRx) which are all write-only.

The 128-bit input data and initialization vector (for some modes) are each stored in four 32-bit registers (AESB_IDATARx and AESB_IVRx) which are all write-only.

As soon as the initialization vector, the input data and the key are configured, the encryption/decryption process may be started. Then the encrypted/decrypted data will be ready to be read out on the four 32-bit output data registers (AESB_ODATARx).

56.2 Embedded Characteristics

- On-the-fly off-chip memory encryption/decryption
- Compliant with *FIPS Publication 197, Advanced Encryption Standard (AES)*
- 128-bit cryptographic key
- On-The-Fly encryption/decryption
- 12 clock cycles encryption/decryption processing time with a 128-bit cryptographic key
- Double input buffer optimizes runtime
- Support of the three standard modes of operation specified in the *NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation - Methods and Techniques:*
  - Electronic Code Book (ECB)
  - Cipher Block Chaining (CBC) including CBC-MAC
  - Counter (CTR)
- Last Output Data mode allows optimized Message Authentication Code (MAC) generation

56.3 Product Dependencies

56.3.1 Power Management

The AESB may be clocked through the Power Management Controller (PMC), so the programmer must first configure the PMC to enable the AESB clock.

56.3.2 Interrupt

The AESB interface has an interrupt line connected to the Interrupt Controller.

Handling the AESB interrupt requires programming the Interrupt Controller before configuring the AESB.

Table 56-1. Peripheral IDs

Instance	ID
AESB	10

56.4 Functional Description

The Advanced Encryption Standard Bridge (AESB) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information.

Encryption converts data to an unintelligible form called ciphertext. Decrypting the ciphertext converts the data back into its original form, called plaintext. The CIPHER bit in the AESB Mode Register (AESB_MR) allows selection between the encryption and the decryption processes.

The AESB is capable of using cryptographic keys of 128 bits to encrypt and decrypt data in blocks of 128 bits. This 128-bit key is defined in the Key Registers (AESB_KEYWRx).

The input to the encryption processes of the CBC mode includes, in addition to the plaintext, a 128-bit data block called the initialization vector (IV), which must be set in the Initialization Vector Registers (AESB_IVRx). The initialization vector is used in an initial step in the encryption of a message and in the corresponding decryption of the message. The Initialization Vector Registers are also used by the CTR mode to set the counter value.

56.4.1 Operating Modes

The AESB supports the following modes of operation:

- ECB—Electronic Code Book
- CBC—Cipher Block Chaining
- CTR—Counter

The data preprocessing, post-processing and data chaining for the operating modes are performed automatically. Refer to the *NIST Special Publication 800-38A Recommendation* for more complete information.

The modes are selected by the OPMOD field in AESB_MR.

In CTR mode, the size of the block counter embedded in the module is 16 bits. Therefore, there is a rollover after processing 1 megabyte of data. If the file to be processed is greater than 1 megabyte, this file must be split into fragments of 1 megabyte or less for the first fragment if the initial value of the counter is greater than 0. Prior to loading the first fragment into AESB_IDATARx registers, the AESB_IVRx registers must be cleared. For any fragment, after the transfer is completed and prior to transferring the next fragment, AESB_IVR0 must be programmed so that the fragment number (0 for the first fragment, 1 for the second one, and so on) is written in the 16 MSB of AESB_IVR0.

If the initial value of the counter is greater than 0 and the data buffer size to be processed is greater than 1 megabyte, the size of the first fragment to be processed must be 1 megabyte minus 16x(initial value) to prevent a rollover of the internal 1-bit counter.

56.4.2 Double Input Buffer

The input data register can be double-buffered to reduce the runtime of large files.

This mode allows writing a new message block when the previous message block is being processed.

The DUALBUFF bit in register AESB_MR must be set to 1 to access the double buffer.

56.4.3 Start Modes

The SMOD field in register AESB_MR allows selection of the Encryption (or Decryption) Start mode.

56.4.3.1 Manual Mode

The sequence is as follows:

1. Write AESB_MR with all required fields, including but not limited to SMOD and OPMOD.
2. Write the 128-bit key in the Key Registers (AESB_KEYWRx).

3. Write the initialization vector (or counter) in the Initialization Vector Registers (AESB_IVRx).

Note: The Initialization Vector Registers concern all modes except ECB.

4. Set the DATRDY (Data Ready) bit in the AESB Interrupt Enable Register (AESB_IER) depending on whether an interrupt is required, or not, at the end of processing.
5. Write the data to be encrypted/decrypted in the authorized Input Data Registers (see Table 56-2).

Table 56-2. Authorized Input Data Registers

Operating Mode	Input Data Registers to Write
ECB	All
CBC	All
CTR	All

6. Set the START bit in the AESB Control Register (AESB_CR) to begin the encryption or decryption process.
7. When processing is complete, the DATRDY bit in the AESB Interrupt Status Register (AESB_ISR) raises. If an interrupt has been enabled by setting the DATRDY bit in AESB_IER, the interrupt line of the AESB is activated.
8. When the software reads one of the Output Data Registers (AESB_ODATARx), the AESB_ISR.DATRDY bit is automatically cleared.

56.4.3.2 Auto Mode

Auto mode is similar to Manual mode, except that in Auto mode, as soon as the correct number of Input Data registers is written, processing starts automatically without any action in the Control Register.

56.4.4 Last Output Data Mode

Last Output Data mode is used to generate cryptographic checksums on data (MAC) by means of a cipher block chaining encryption algorithm (the CBC-MAC algorithm for example).

After each end of encryption/decryption, the output data are available on the output data registers for Manual and Auto modes.

The Last Output Data (LOD) bit in AESB_MR allows retrieval of only the last data of several encryption/decryption processes.

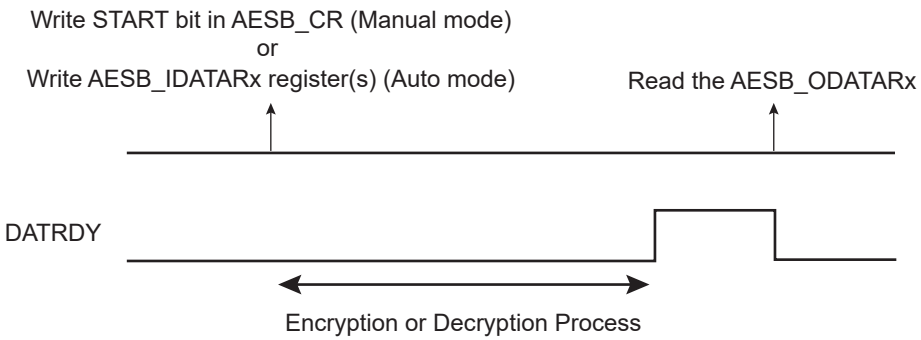
Those data are only available on the Output Data Registers (AESB_ODATARx).

56.4.5 Manual and Auto Modes

56.4.5.1 If AESB_MR.LOD = 0

The AESB_ISR.DATRDY bit is cleared when at least one of the Output Data Registers is read (see Figure 56-1).

Figure 56-1. Manual and Auto Modes with AESB_MR.LOD = 0

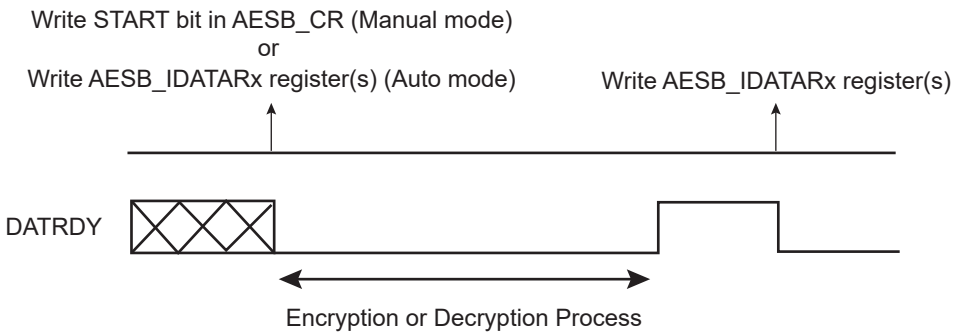


If the user does not want to read the output data registers between each encryption/decryption, the AESB_ISR.DATRDY bit will not be cleared. If the AESB_ISR.DATRDY bit is not cleared, the user cannot know the end of the following encryptions/decryptions.

56.4.5.2 If AESB_MR.LOD = 1

The AESB_ISR.DATRDY bit is cleared when at least one Input Data Register is written, so before the start of a new transfer (see Figure 56-2). No more Output Data Register reads are necessary between consecutive encryptions/decryptions.

Figure 56-2. Manual and Auto Modes with AESB_MR.LOD = 1



56.4.6 Automatic Bridge Mode

56.4.6.1 Description

The Automatic Bridge mode, when the AESB block is connected between the system bus and a DDR port, provides automatic encryption/decryption to/from a DDR port without any action on the part of the user. For Automatic Bridge mode, the OPMODE field must be configured to 0x4 in AESB_MR (see Section 56.6.2 “AESB Mode Register”). If bit AESB_MR.AAHB is set and field AESB_MR.OPMODE = 0x4, there is no compliance with the standard CTR mode of operation.

In case of write transfer, this mode automatically encrypts the data before writing it to the final slave destination. In case of read transfer, this mode automatically decrypts the data read from the target slave before putting it on the system bus.

Therefore, this mode does not work if the automatically encrypted data is moved at another address outside of the AESB IP scope. This means that for a given data, the encrypted value is not the same if written at different addresses.

56.4.6.2 Configuration

The Automatic Bridge mode can be enabled by setting bit AESB_MR.AAHB.

The IV (Initialization Vector) field of the AESB Initialization Vector Register x (AESB_IVRx) can be used to add a nonce in the encryption process in order to bring even more security (ignored if not filled). In this case, any value encrypted with a given nonce can only be decrypted with this nonce. If another nonce is set for the IV field, any value encrypted with the previous nonce cannot be decrypted anymore (see Section 56.6.10 “AESB Initialization Vector Register x”).

Dual buffer usage (write a 1 to bit AESB_MR.DUALBUFF) is recommended for improved performance.

56.5 Security Features

56.5.1 Unspecified Register Access Detection

When an unspecified register access occurs, the URAD bit in AESB_ISR raises. Its source is then reported in the Unspecified Register Access Type (URAT) field. Only the last unspecified register access is available through the URAT field.

Several kinds of unspecified register accesses can occur:

- Input Data Register written during the data processing when SMOD = IDATAR0_START
- Output Data Register read during data processing
- Mode Register written during data processing
- Output Data Register read during subkeys generation
- Mode Register written during subkeys generation
- Write-only register read access

The URAD bit and the URAT field can only be reset by the SWRST bit in AESB_CR.

56.6 Advanced Encryption Standard Bridge (AESB) User Interface

Table 56-3. Register Mapping

Offset	Register	Name	Access	Reset
0x00	Control Register	AESB_CR	Write-only	–
0x04	Mode Register	AESB_MR	Read/Write	0x0
0x08–0x0C	Reserved	–	–	–
0x10	Interrupt Enable Register	AESB_IER	Write-only	–
0x14	Interrupt Disable Register	AESB_IDR	Write-only	–
0x18	Interrupt Mask Register	AESB_IMR	Read-only	0x0
0x1C	Interrupt Status Register	AESB_ISR	Read-only	0x0
0x20	Key Word Register 0	AESB_KEYWR0	Write-only	–
0x24	Key Word Register 1	AESB_KEYWR1	Write-only	–
0x28	Key Word Register 2	AESB_KEYWR2	Write-only	–
0x2C	Key Word Register 3	AESB_KEYWR3	Write-only	–
0x30–0x3C	Reserved	–	–	–
0x40	Input Data Register 0	AESB_IDATAR0	Write-only	–
0x44	Input Data Register 1	AESB_IDATAR1	Write-only	–
0x48	Input Data Register 2	AESB_IDATAR2	Write-only	–
0x4C	Input Data Register 3	AESB_IDATAR3	Write-only	–
0x50	Output Data Register 0	AESB_ODATAR0	Read-only	0x0
0x54	Output Data Register 1	AESB_ODATAR1	Read-only	0x0
0x58	Output Data Register 2	AESB_ODATAR2	Read-only	0x0
0x5C	Output Data Register 3	AESB_ODATAR3	Read-only	0x0
0x60	Initialization Vector Register 0	AESB_IVR0	Write-only	–
0x64	Initialization Vector Register 1	AESB_IVR1	Write-only	–
0x68	Initialization Vector Register 2	AESB_IVR2	Write-only	–
0x6C	Initialization Vector Register 3	AESB_IVR3	Write-only	–
0x70–0xFC	Reserved	–	–	–

56.6.1 AESB Control Register

Name: AESB_CR

Address: 0xF001C000

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	SWRST
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	START

- **START: Start Processing**

0: No effect

1: Starts manual encryption/decryption process

- **SWRST: Software Reset**

0: No effect

1: Resets the AESB. A software triggered hardware reset of the AESB interface is performed.

56.6.2 AESB Mode Register

Name: AESB_MR

Address: 0xF001C004

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
CKEY				–	–	–	–
15	14	13	12	11	10	9	8
LOD	OPMOD			–	–	SMOD	
7	6	5	4	3	2	1	0
PROCDLY				DUALBUFF	AAHB	–	CIPHER

- **CIPHER: Processing Mode**

0: Decrypts data

1: Encrypts data

- **AAHB: Automatic Bridge Mode**

0: Automatic Bridge mode disabled

1: Automatic Bridge mode enabled

- **DUALBUFF: Dual Input Buffer**

Value	Name	Description
0x0	INACTIVE	AESB_IDATARx cannot be written during processing of previous block.
0x1	ACTIVE	AESB_IDATARx can be written during processing of previous block when SMOD = 0x2. It speeds up the overall runtime of large files.

- **PROCDLY: Processing Delay**

Processing Time = 12 × (PROCDLY + 1)

The Processing Time represents the number of clock cycles that the AESB needs in order to perform one encryption/decryption .

Note: The best performance is achieved with PROCDLY equal to 0.

- **SMOD: Start Mode**

Value	Name	Description
0x0	MANUAL_START	Manual mode
0x1	AUTO_START	Auto mode
0x2	IDATAR0_START	AESB_IDATAR0 access only Auto mode

Values which are not listed in the table must be considered as “reserved”.

- **OPMOD: Operating Mode**

Value	Name	Description
0x0	ECB	Electronic Code Book mode
0x1	CBC	Cipher Block Chaining mode
0x2	–	Reserved
0x3	–	Reserved
0x4	CTR	Counter mode (16-bit internal counter)

Values which are not listed in the table must be considered as “reserved”.

For CBC-MAC operating mode, configure OPMOD to 0x1 (CBC) and set LOD to 1.

Note: If the OPMODE field is set to 0x4 and AAHB = 1, there is no compliance with the standard CTR mode of operation.

- **LOD: Last Output Data Mode**

0: No effect.

After each end of encryption/decryption, the output data will be available either on the output data registers (Manual and Auto modes).

In Manual and Auto modes, the AESB_ISR.DATRDY bit is cleared when at least one of the Output Data registers is read.

1: The AESB_ISR.DATRDY bit is cleared when at least one of the Input Data Registers is written.

No more Output Data Register reads are necessary between consecutive encryptions/decryptions (see Section 56.4.4 “Last Output Data Mode”).

- **CKEY: Key**

Value	Name	Description
0xE	PASSWD	This field must be written with 0xE the first time that AES_MR is programmed. For subsequent programming of the AES_MR register, any value can be written, including that of 0xE.Always reads as 0.

56.6.3 AESB Interrupt Enable Register

Name: AESB_IER

Address: 0xF001C010

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	URAD
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

- **DATRDY: Data Ready Interrupt Enable**
- **URAD: Unspecified Register Access Detection Interrupt Enable**

56.6.4 AESB Interrupt Disable Register

Name: AESB_IDR

Address: 0xF001C014

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	URAD
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

- **DATRDY: Data Ready Interrupt Disable**
- **URAD: Unspecified Register Access Detection Interrupt Disable**

56.6.5 AESB Interrupt Mask Register

Name: AESB_IMR

Address: 0xF001C018

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	URAD
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

- **DATRDY: Data Ready Interrupt Mask**
- **URAD: Unspecified Register Access Detection Interrupt Mask**

56.6.6 AESB Interrupt Status Register

Name: AESB_ISR

Address: 0xF001C01C

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
URAT				–	–	–	URAD
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

- **DATRDY: Data Ready**

0: Output data not valid.

1: Encryption or decryption process is completed.

DATRDY is cleared when a Manual encryption/decryption occurs (START bit in AESB_CR) or when a software triggered hardware reset of the AESB interface is performed (SWRST bit in AESB_CR).

AESB_MR.LOD = 0: In Manual and Auto modes, the DATRDY bit can also be cleared when at least one of the Output Data Registers is read.

AESB_MR.LOD = 1: In Manual and Auto modes, the DATRDY bit can also be cleared when at least one of the Input Data Registers is written.

- **URAD: Unspecified Register Access Detection Status**

0: No unspecified register access has been detected since the last SWRST.

1: At least one unspecified register access has been detected since the last SWRST.

URAD bit is reset only by the SWRST bit in AESB_CR.

- **URAT: Unspecified Register Access**

Value	Name	Description
0x0	IDR_WR_PROCESSING	Input Data Register written during the data processing when SMOD = 0x2 mode
0x1	ODR_RD_PROCESSING	Output Data Register read during the data processing
0x2	MR_WR_PROCESSING	Mode Register written during the data processing
0x3	ODR_RD_SUBKGEN	Output Data Register read during the subkeys generation
0x4	MR_WR_SUBKGEN	Mode Register written during the subkeys generation
0x5	WOR_RD_ACCESS	Write-only register read access

Only the last Unspecified Register Access Type is available through the URAT field.

URAT field is reset only by the SWRST bit in AESB_CR.

56.6.7 AESB Key Word Register x

Name: AESB_KEYWRx

Address: 0xF001C020

Access: Write-only

31	30	29	28	27	26	25	24
KEYW							
23	22	21	20	19	18	17	16
KEYW							
15	14	13	12	11	10	9	8
KEYW							
7	6	5	4	3	2	1	0
KEYW							

- **KEYW: Key Word**

The four 32-bit Key Word registers set the 128-bit cryptographic key used for encryption/decryption.

AESB_KEYWR0 corresponds to the first word of the key, AESB_KEYWR3 to the last one.

These registers are write-only to prevent the key from being read by another application.

56.6.8 AESB Input Data Register x

Name: AESB_IDATARx

Address: 0xF001C040

Access: Write-only

31	30	29	28	27	26	25	24
IDATA							
23	22	21	20	19	18	17	16
IDATA							
15	14	13	12	11	10	9	8
IDATA							
7	6	5	4	3	2	1	0
IDATA							

- **IDATA: Input Data Word**

The four 32-bit Input Data registers set the 128-bit data block used for encryption/decryption.

AESB_IDATAR0 corresponds to the first word of the data to be encrypted/decrypted, AESB_IDATAR3 to the last one.

These registers are write-only to prevent the input data from being read by another application.

56.6.9 AESB Output Data Register x

Name: AESB_ODATARx

Address: 0xF001C050

Access: Read-only

31	30	29	28	27	26	25	24
ODATA							

23	22	21	20	19	18	17	16
ODATA							

15	14	13	12	11	10	9	8
ODATA							

7	6	5	4	3	2	1	0
ODATA							

- **ODATA: Output Data**

The four 32-bit Output Data registers contain the 128-bit data block that has been encrypted/decrypted.

AESB_ODATAR0 corresponds to the first word, AESB_ODATAR3 to the last one.

56.6.10 AESB Initialization Vector Register x

Name: AESB_IVRx

Address: 0xF001C060

Access: Write-only

31	30	29	28	27	26	25	24
IV							

23	22	21	20	19	18	17	16
IV							

15	14	13	12	11	10	9	8
IV							

7	6	5	4	3	2	1	0
IV							

- **IV: Initialization Vector**

The four 32-bit Initialization Vector registers set the 128-bit Initialization Vector data block that is used by some modes of operation as an additional initial input.

AESB_IVR0 corresponds to the first word of the Initialization Vector, AESB_IVR3 to the last one.

These registers are write-only to prevent the Initialization Vector from being read by another application.

For CBC mode, the IV input value corresponds to the initialization vector.

For CTR mode, the IV input value corresponds to the initial counter value.

Note: These registers are not used in ECB mode and must not be written. For Automatic Bridge dedicated mode, the IV input value corresponds to the initial nonce.

57. Advanced Encryption Standard (AES)

57.1 Description

The Advanced Encryption Standard (AES) is compliant with the American *FIPS (Federal Information Processing Standard) Publication 197* specification.

The AES supports all five confidentiality modes of operation for symmetrical key block cipher algorithms (ECB, CBC, OFB, CFB and CTR), as specified in the *NIST Special Publication 800-38A Recommendation,* as well as Galois/Counter Mode (GCM) as specified in the *NIST Special Publication 800-38D Recommendation*. It is compatible with all these modes via DMA Controller channels, minimizing processor intervention for large buffer transfers.

The 128-bit/192-bit/256-bit key is stored in four/six/eight 32-bit write-only AES Key Word registers (AES_KEYWR0–3).

The 128-bit input data and initialization vector (for some modes) are each stored in four 32-bit write-only AES Input Data registers (AES_IDATAR0–3) and AES Initialization Vector registers (AES_IVR0–3).

As soon as the initialization vector, the input data and the key are configured, the encryption/decryption process may be started. Then the encrypted/decrypted data are ready to be read out on the four 32-bit AES Output Data registers (AES_ODATAR0–3) or through the DMA channels.

57.2 Embedded Characteristics

- Compliant with *FIPS Publication 197, Advanced Encryption Standard (AES)*
- 128-bit/192-bit/256-bit Cryptographic Key
- 12/14/16 Clock Cycles Encryption/Decryption Processing Time with a 128-bit/192-bit/256-bit Cryptographic Key
- Double Input Buffer Optimizes Runtime
- Automatic Padding supported for IPSEC and SSL standards
- IPSEC and SSL Protocol Layers Improved Performances (Tightly coupled with SHA)
- Support of the Modes of Operation Specified in the *NIST Special Publication 800-38A* and *NIST Special Publication 800-38D*:
  - Electronic Code Book (ECB)
  - Cipher Block Chaining (CBC) including CBC-MAC
  - Cipher Feedback (CFB)
  - Output Feedback (OFB)
  - Counter (CTR)
  - Galois/Counter Mode (GCM)
  - XEX-Based Tweaked-Codebook Mode (XTS)
- 8, 16, 32, 64 and 128-bit Data Sizes Possible in CFB Mode
- Last Output Data Mode Allows Optimized Message Authentication Code (MAC) Generation
- Connection to DMA Optimizes Data Transfers for all Operating Modes

57.3 Product Dependencies

57.3.1 Power Management

The AES may be clocked through the Power Management Controller (PMC), so the programmer must first to configure the PMC to enable the AES clock.

57.3.2 Interrupt Sources

The AES interface has an interrupt line connected to the Interrupt Controller.

Handling the AES interrupt requires programming the Interrupt Controller before configuring the AES.

Table 57-1. Peripheral IDs

Instance	ID
AES	9

57.4 Functional Description

The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information.

Encryption converts data to an unintelligible form called ciphertext. Decrypting the ciphertext converts the data back into its original form, called plaintext. The CIPHER bit in the AES Mode register (AES_MR) allows selection between the encryption and the decryption processes.

The AES is capable of using cryptographic keys of 128/192/256 bits to encrypt and decrypt data in blocks of 128 bits. This 128-bit/192-bit/256-bit key is defined in the AES_KEYWRx.

The input to the encryption processes of the CBC, CFB, and OFB modes includes, in addition to the plaintext, a 128-bit data block called the initialization vector (IV), which must be set in the AES_IVRx. The initialization vector is used in an initial step in the encryption of a message and in the corresponding decryption of the message. The AES_IVRx are also used by the CTR mode to set the counter value.

57.4.1 AES Register Endianness

In ARM processor-based products, the system bus and processors manipulate data in little-endian form. The AES interface requires little-endian format words. However, in accordance with the protocol of the FIPS 197 specification, data is collected, processed and stored by the AES algorithm in big-endian form.

The following example illustrates how to configure the AES:

If the first 64 bits of a message (according to FIPS 197, i.e., big-endian format) to be processed is 0xcafedeca_01234567, then the AES_IDATAR0 and AES_IDATAR1 registers must be written with the following pattern:

- AES_IDATAR0 = 0xcadefeca
- AES_IDATAR1 = 0x67452301

57.4.2 Operating Modes

The AES supports the following modes of operation:

- ECB: Electronic Code Book
- CBC: Cipher Block Chaining
- OFB: Output Feedback
- CFB: Cipher Feedback
  - CFB8 (CFB where the length of the data segment is 8 bits)
  - CFB16 (CFB where the length of the data segment is 16 bits)
  - CFB32 (CFB where the length of the data segment is 32 bits)
  - CFB64 (CFB where the length of the data segment is 64 bits)
  - CFB128 (CFB where the length of the data segment is 128 bits)
- CTR: Counter
- GCM: Galois/Counter Mode

The data preprocessing, data post-processing and data chaining for the concerned modes are performed automatically. Refer to the *NIST Special Publication 800-38A* and *NIST Special Publication 800-38D* for more complete information.

These modes are selected by setting the OPMOD field in the AES_MR.

In CFB mode, five data sizes are possible (8, 16, 32, 64 or 128 bits), configurable by means of the CFBS field in the AES_MR (Section 57.5.2 “AES Mode Register”).

In CTR mode, the size of the block counter embedded in the module is 16 bits. Therefore, there is a rollover after processing 1 megabyte of data. If the file to be processed is greater than 1 megabyte, this file must be split into fragments of 1 megabyte or less for the first fragment if the initial value of the counter is greater than 0. Prior to loading the first fragment into AES_IDATARx, AES_IVRx must be fully programmed with the initial counter value. For any fragment, after the transfer is completed and prior to transferring the next fragment, AES_IVRx must be programmed with the appropriate counter value.

57.4.3 Double Input Buffer

The AES_IDATARx can be double-buffered to reduce the runtime of large files.

This mode allows writing a new message block when the previous message block is being processed. This is only possible when DMA accesses are performed (SMOD = 0x2).

The DUALBUFF bit in the AES_MR must be set to ‘1’ to access the double buffer.

57.4.4 Start Modes

The SMOD field in the AES_MR allows selection of the encryption (or decryption) Start mode.

57.4.4.1 Manual Mode

The sequence of actions is as follows:

1. Write the AES_MR with all required fields, including but not limited to SMOD and OPMOD.
2. Write the 128-bit/192-bit/256-bit key in the AES_KEYWRx.
3. Write the initialization vector (or counter) in the AES_IVRx.

Note: The AES_IVRx concerns all modes except ECB.

4. Set the bit DATRDY (Data Ready) in the AES Interrupt Enable register (AES_IER), depending on whether an interrupt is required or not at the end of processing.
5. Write the data to be encrypted/decrypted in the authorized AES_IDATARx (see Table 57-2).
6. Set the START bit in the AES Control register (AES_CR) to begin the encryption or the decryption process.

7. When processing completes, the DATRDY flag in the AES Interrupt Status register (AES_ISR) is raised. If an interrupt has been enabled by setting the DATRDY bit in the AES_IER, the interrupt line of the AES is activated.
8. When software reads one of the AES_ODATARx, the DATRDY bit is automatically cleared.

Table 57-2. Authorized Input Data Registers

Operating Mode	Input Data Registers to Write
ECB	All
CBC	All
OFB	All
128-bit CFB	All
64-bit CFB	AES_IDATAR0 and AES_IDATAR1
32-bit CFB	AES_IDATAR0
16-bit CFB	AES_IDATAR0
8-bit CFB	AES_IDATAR0
CTR	All
GCM	All

Notes: 1. In 64-bit CFB mode, writing to AES_IDATAR2 and AES_IDATAR3 is not allowed and may lead to errors in processing.
2. In 32, 16, and 8-bit CFB modes, writing to AES_IDATAR1, AES_IDATAR2 and AES_IDATAR3 is not allowed and may lead to errors in processing.

57.4.4.2 Auto Mode

The Auto Mode is similar to the manual one, except that in this mode, as soon as the correct number of AES_IDATARx is written, processing is automatically started without any action in the AES_CR.

57.4.4.3 DMA Mode

The DMA Controller can be used in association with the AES to perform an encryption/decryption of a buffer without any action by software during processing.

The SMOD field in the AES_MR must be configured to 0x2 and the DMA must be configured with non-incremental addresses.

The start address of any transfer descriptor must be configured with the address of AES_IDATAR0.

The DMA chunk size configuration depends on the AES mode of operation and is listed in Table 57-3 “DMA Data Transfer Type for the Different Operating Modes”.

When writing data to AES with a first DMA channel, data are first fetched from a memory buffer (source data). It is recommended to configure the size of source data to “words” even for CFB modes. On the contrary, the destination data size depends on the mode of operation. When reading data from the AES with the second DMA channel, the source data is the data read from AES and data destination is the memory buffer. In this case, the source data size depends on the AES mode of operation and is listed in Table 57-3.

Table 57-3. DMA Data Transfer Type for the Different Operating Modes

Operating Mode	Chunk Size	Destination/Source Data Transfer Type
ECB	4	Word
CBC	4	Word
OFB	4	Word
CFB 128-bit	4	Word
CFB 64-bit	1	Word
CFB 32-bit	1	Word
CFB 16-bit	1	Half-word
CFB 8-bit	1	Byte
CTR	4	Word
GCM	4	Word

57.4.5 Last Output Data Mode

This mode is used to generate cryptographic checksums on data (MAC) by means of cipher block chaining encryption algorithm (CBC-MAC algorithm for example).

After each end of encryption/decryption, the output data are available either on the AES_ODATARx for Manual and Auto mode or at the address specified in the receive buffer pointer for DMA mode (see Table 57-4).

The Last Output Data (LOD) bit in the AES_MR allows retrieval of only the last data of several encryption/decryption processes.

Therefore, there is no need to define a read buffer in DMA mode.

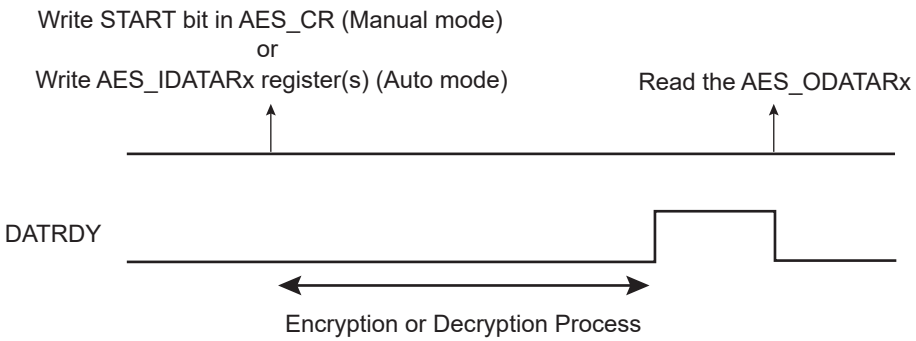
This data are only available on the AES_ODATARx.

57.4.5.1 Manual and Auto Modes

If AES_MR.LOD = 0

The DATRDY flag is cleared when at least one of the AES_ODATARx is read (see Figure 57-1).

Figure 57-1. Manual and Auto Modes with AES_MR.LOD = 0



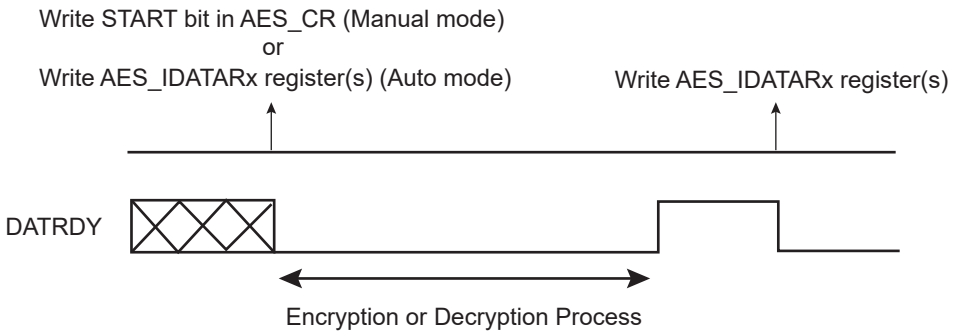
If the user does not want to read the AES_ODATARx between each encryption/decryption, the DATRDY flag will not be cleared. If the DATRDY flag is not cleared, the user cannot know the end of the following encryptions/decryptions.

If AES_MR.LOD = 1

This mode is optimized to process AES CPC-MAC operating mode.

The DATRDY flag is cleared when at least one AES_IDATAR is written (see Figure 57-2). No more AES_ODATAR reads are necessary between consecutive encryptions/decryptions.

Figure 57-2. Manual and Auto Modes with AES_MR.LOD = 1



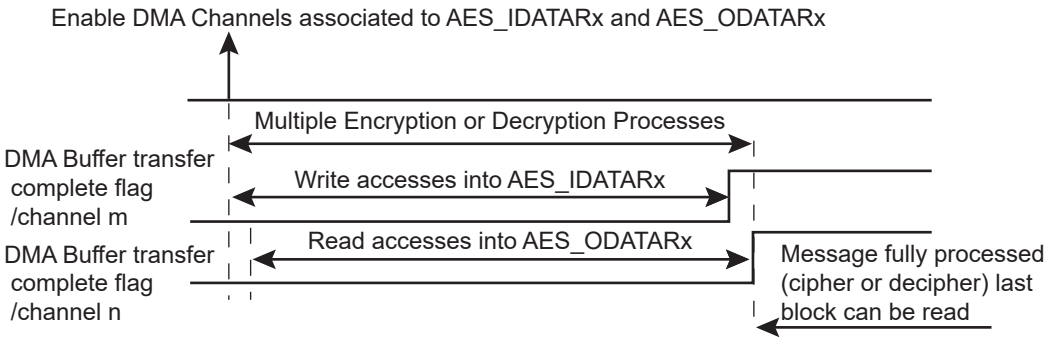
57.4.5.2 DMA Mode

If AES_MR.LOD = 0

This mode may be used for all AES operating modes except CBC-MAC where AES_MR.LOD = 1 mode is recommended.

The end of the encryption/decryption is indicated by the end of DMA transfer associated to AES_ODATARx (see Figure 57-3). Two DMA channels are required: one for writing message blocks to AES_IDATARx and one to obtain the result from AES_ODATARx.

Figure 57-3. DMA Transfer with AES_MR.LOD = 0



If AES_MR.LOD = 1

This mode is optimized to process AES CBC-MAC operating mode.

The user must first wait for the DMA buffer transfer complete flag, then for the flag DATRDY to rise to ensure that the encryption/decryption is completed (see Figure 57-4).

In this case, no receive buffers are required.

The output data are only available on the AES_ODATARx.

Figure 57-4. DMA Transfer with AES_MR.LOD = 1

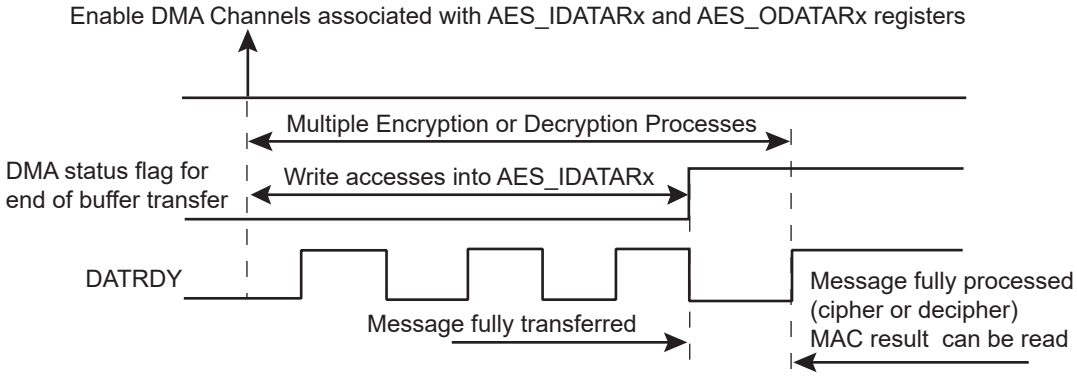


Table 57-4 summarizes the different cases.

Table 57-4. Last Output Data Mode Behavior versus Start Modes

Sequence	Manual and Auto Modes		DMA Transfer	
	AES_MR.LOD = 0	AES_MR.LOD = 1	AES_MR.LOD = 0	AES_MR.LOD = 1
DATRDY Flag Clearing Condition(1)	At least one AES_ODATAR must be read	At least one AES_IDATAR must be written	Not used	Managed by the DMA
End of Encryption/Decryption Notification	DATRDY	DATRDY	2 DMA Buffer transfer complete flags (channel m and channel n)	DMA buffer transfer complete flag, then AES DATRDY flag
Encrypted/Decrypted Data Result Location	In the AES_ODATARx	In the AES_ODATARx	At the address specified in the Channel Buffer Transfer Descriptor	In the AES_ODATARx

Note: 1. Depending on the mode, there are other ways of clearing the DATRDY flag. See Section 57.5.6 “AES Interrupt Status Register”.

Warning: In DMA mode, reading the AES_ODATARx before the last data transfer may lead to unpredictable results.

57.4.6 Galois/Counter Mode (GCM)

57.4.6.1 Description

GCM comprises the AES engine in CTR mode along with a universal hash function (GHASH engine) that is defined over a binary Galois field to produce a message authentication tag (the AES CTR engine and the GHASH engine are depicted in Figure 57-5).

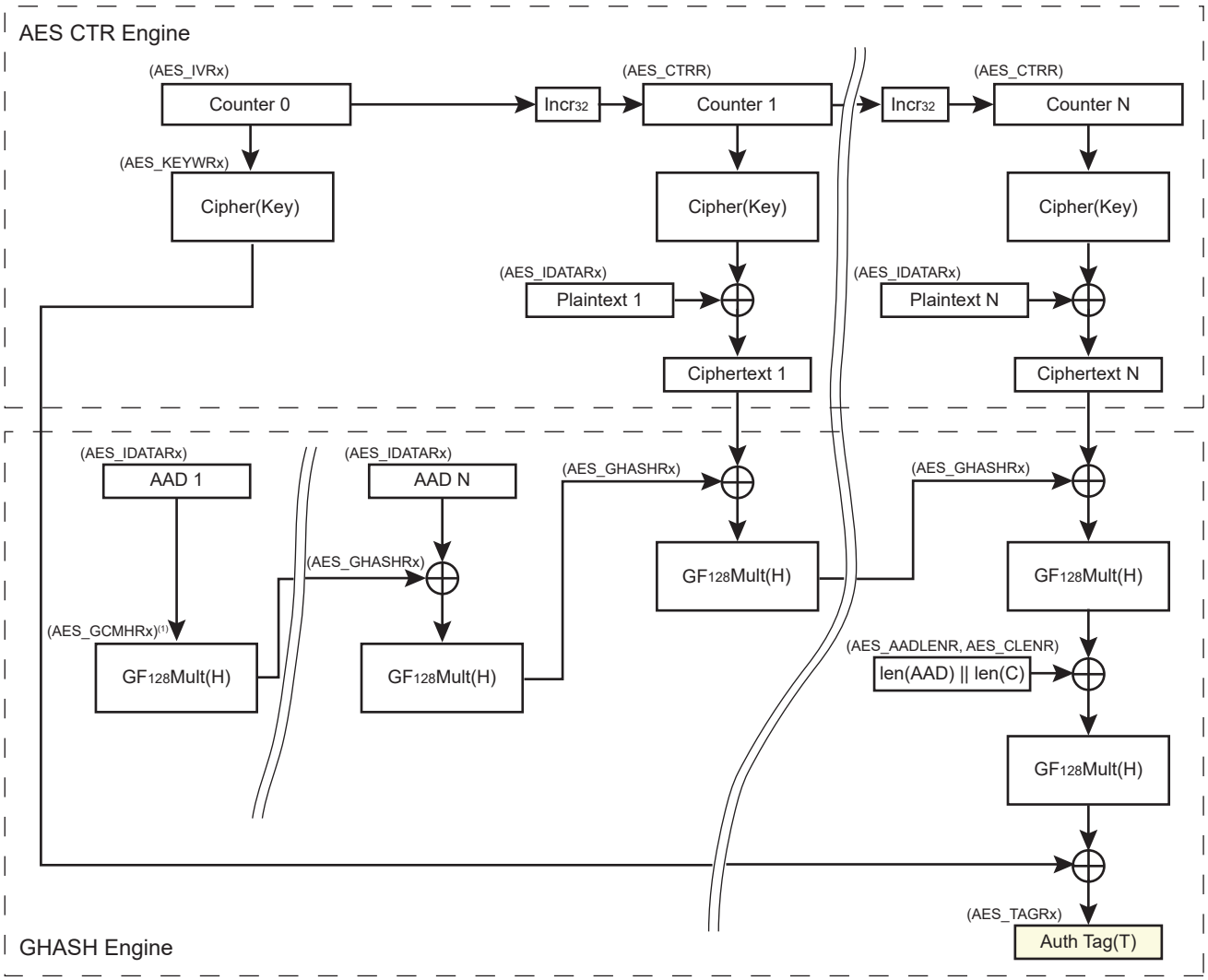
The GHASH engine processes data packets after the AES operation. GCM provides assurance of the confidentiality of data through the AES Counter mode of operation for encryption. Authenticity of the confidential data is assured through the GHASH engine. GCM can also provide assurance of data that is not encrypted. Refer to the *NIST Special Publication 800-38D* for more complete information.

GCM can be used with or without the DMA master. Messages may be processed as a single complete packet of data or they may be broken into multiple packets of data over time.

GCM processing is computed on 128-bit input data fields. There is no support for unaligned data. The AES key length can be whatever length is supported by the AES module.

The recommended programming procedure when using DMA is described in Section 57.4.6.3.

Figure 57-5. GCM Block Diagram



Note: 1. Optional

57.4.6.2 Key Writing and Automatic Hash Subkey Calculation

Whenever a new key (AES_KEYWRx) is written to the hardware, two automatic actions are processed:

- GCM Hash Subkey *H* generation—The GCM hash subkey (*H*) is automatically generated. The GCM hash subkey generation must be complete before doing any other action. The DATRDY bit of the AES_ISR indicates when the subkey generation is complete (with interrupt if configured). The GCM hash subkey calculation is processed with the formula *H* = CIPHER(Key, <128 bits to zero>). The generated GCM *H* value is then available in the AES_GCMHRx. If the application software requires a specific hash subkey, the automatically generated H value can be overwritten in the AES_GCMHRx.
  The AES_GCMHRx can be written after the end of the hash subkey generation (see AES_ISR.DATRDY) and prior to starting the input data feed.
- AES_GHASHRx Clear—The AES_GHASHRx are automatically cleared. If a hash initial value is needed for the GHASH, it must be written to the AES_GHASHRx
  - after a write to AES_KEYWRx, if any
  - before starting the input data feed

57.4.6.3 GCM Processing

GCM processing comprises three phases:

1. Processing the Additional Authenticated Data (*AAD*), hash computation only.
2. Processing the Ciphertext (*C*), hash computation + ciphering/deciphering.
3. Generating the Tag using length of *AAD*, length of *C* and J_0 (see NIST documentation for details).

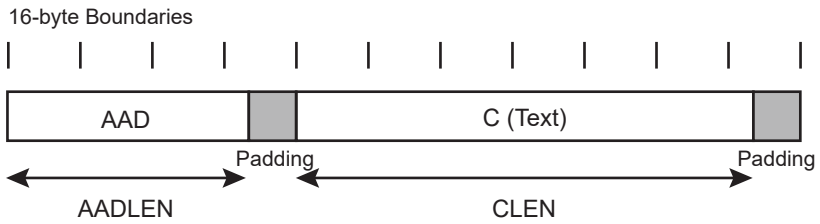
The Tag generation can be done either automatically, after the end of *AAD*/*C* processing if TAG_EN bit is set in the AES_MR or done manually, using the GHASH field in AES_GHASHRx (See subsections "Processing a Complete Message with Tag Generation" and "Manual GCM Tag Generation" below for details).

Processing a Complete Message with Tag Generation

Use this procedure only if J_0 four LSB bytes ≠ 0xFFFFFFFF.

NOTE: In the case where J_0 four LSB bytes = 0xFFFFFFFF or if the value is unknown, use the procedure described in "Processing a Complete Message without Tag Generation" followed by the procedure in "Manual GCM Tag Generation".

Figure 57-6. Full Message Alignment



To process a complete message with Tag generation, the sequence is as follows:

1. In AES_MR set OPMOD to GCM and GTAGEN to '1' (configuration as usual for the rest).
2. Set KEYW in AES_KEYWRx and wait until DATRDY bit of AES_ISR is set (GCM hash subkey generation complete); use interrupt if needed. See Section 57.4.6.2 "Key Writing and Automatic Hash Subkey Calculation".
3. Calculate the J_0 value as described in NIST documentation $J_0 = IV \,||\, 0^{31} \,||\, 1$ when len(IV) = 96 and $J_0 = \text{GHASH}_H(IV \,||\, 0^{s+64} \,||\, [\text{len}(IV)]_{64})$ if len(IV) ≠ 96. See "Processing a Message with only AAD (GHASHH)" for J_0 generation.
4. Set IV in AES_IVRx with inc32(J_0) (J_0 + 1 on 32 bits).
5. Set AADLEN field in AES_AADLENR and CLEN field in AES_CLENR.
6. Fill the IDATA field of AES_IDATARx with the message to process according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Wait for TAGRDY to be set (use interrupt if needed), then read the TAG field of AES_TAGRx to obtain the authentication tag of the message.

Processing a Complete Message without Tag Generation

Processing a message without generating the Tag can be used to customize the Tag generation, or to process a fragmented message. To manually generate the GCM Tag, refer to "Manual GCM Tag Generation".

To process a complete message without Tag generation, the sequence is as follows:

1. In AES_MR set OPMOD to GCM and GTAGEN to '0' (configuration as usual for the rest).
2. Set KEYW in AES_KEYWRx and wait until DATRDY bit of AES_ISR is set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in the AES_GCMHRx. See Section 57.4.6.2 "Key Writing and Automatic Hash Subkey Calculation".

3. Calculate the J_0 value as described in NIST documentation $J_0 = IV \| 0^{31} \| 1$ when len(IV) = 96 and $J_0 = \text{GHASH}_H(IV \| 0^{s+64} \| [\text{len}(IV)]_{64})$ if len(IV) ≠ 96. See "Processing a Message with only AAD (GHASHH)" for J_0 generation example when len(IV) ≠ 96.
4. Set IV in AES_IVRx with inc32(J_0) (J_0 + 1 on 32 bits).
5. Set AADLEN field in AES_AADLENR and CLEN field in AES_CLENR.
6. Fill the IDATA field of AES_IDATARx with the message to process according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Make sure the last output data have been read if CLEN ≠ 0 (or wait for DATRDY), then read the GHASH field of AES_GHASHRx to obtain the hash value after the last processed data.

Processing a Fragmented Message without Tag Generation

If needed, a message can be processed by fragments, in such case automatic GCM Tag generation is not supported.

To process a message by fragments, the sequence is as follows:

- First fragment:

1. In AES_MR set OPMOD to GCM and GTAGEN to '0' (configuration as usual for the rest).
2. Set KEYW in AES_KEYWRx and wait for DATRDY bit of AES_ISR to be set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in the AES_GCMHRx. See Section 57.4.6.2 "Key Writing and Automatic Hash Subkey Calculation".
3. Calculate the J_0 value as described in NIST documentation $J_0 = IV \| 0^{31} \| 1$ when len(IV) = 96 and $J_0 = \text{GHASH}_H(IV \| 0^{s+64} \| [\text{len}(IV)]_{64})$ if len(IV) ≠ 96. See "Processing a Message with only AAD (GHASHH)" for J_0 generation example when len(IV) ≠ 96.
4. Set IV in AES_IVRx with inc32(J_0) (J_0 + 1 on 32 bits).
5. Set AADLEN field in AES_AADLENR and CLEN field in AES_CLENR according to the length of the first fragment, or set the fields with the full message length, both configurations work.
6. Fill the IDATA field of AES_IDATARx with the first fragment of the message to process (aligned on 16-byte boundary) according to the SMOD configuration used. If Manual Mode or Auto Mode is used the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Make sure the last output data have been read if the fragment ends in C phase (or wait for DATRDY if the fragment ends in *AAD* phase), then read the GHASH field of AES_GHASHRx to obtain the value of the hash after the last processed data and finally read the CTR field of the AES_CTR to obtain the value of the CTR encryption counter (not needed when the fragment ends in *AAD* phase).

- Next fragment (or last fragment):

1. In AES_MR set OPMOD to GCM and GTAGEN to '0' (configuration as usual for the rest).
2. Set KEYW in AES_KEYWRx and wait until DATRDY bit of AES_ISR is set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in the AES_GCMHRx. See Section 57.4.6.2 "Key Writing and Automatic Hash Subkey Calculation".
3. Set IV in AES_IVRx with:
   - If the first block of the fragment is a block of Additional Authenticated data, set IV in AES_IVRx with the J0 initial value
   - If the first block of the fragment is a block of Plaintext data, set IV in AES_IVRx with a value constructed as follows: 'LSB96(J0) || CTR' value, (96 bit LSB of J0 concatenated with saved CTR value from previous fragment).

4. Set AADLEN field in AES_AADLENR and CLEN field in AES_CLENR according to the length of the current fragment, or set the fields with the remaining message length, both configurations work.
5. Fill the GHASH field of AES_GHASHRx with the value stored after the previous fragment.
6. Fill the IDATA field of AES_IDATARx with the current fragment of the message to process (aligned on 16 byte boundary) according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Make sure the last output data have been read if the fragment ends in C phase (or wait for DATRDY if the fragment ends in *AAD* phase), then read the GHASH field of AES_GHASHRx to obtain the value of the hash after the last processed data and finally read the CTR field of the AES_CTR to obtain the value of the CTR encryption counter (not needed when the fragment ends in *AAD* phase).

Note: Step 1 and 2 are required only if the value of the concerned registers has been modified.

Once the last fragment has been processed, the GHASH value will allow manual generation of the GCM tag. See "Manual GCM Tag Generation".

Manual GCM Tag Generation

This section describes the last steps of the GCM Tag generation.

The Manual GCM Tag Generation is used to complete the GCM Tag Generation when the message has been processed without Tag Generation.

Note: The Message Processing without Tag Generation must be finished before processing the Manual GCM Tag Generation.

To generate a GCM Tag manually, the sequence is as follows:

Processing $S = \text{GHASH}_H(AAD \,||\, 0_v \,||\, C \,||\, 0_u \,||\, [\text{len}(AAD)]_{64} \,||\, [\text{len}(C)]_{64})$:

1. In AES_MR set OPMOD to GCM and GTAGEN to '0' (configuration as usual for the rest).
2. Set KEYW in AES_KEYWRx and wait for DATRDY bit of AES_ISR to be set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in the AES_GCMHRx. See Section 57.4.6.2 "Key Writing and Automatic Hash Subkey Calculation".
3. Set AADLEN field to 0x10 (16 bytes) in AES_AADLENR and CLEN field to '0' in AES_CLENR. This will allow running a single GHASH_H on a 16-byte input data (see Figure 57-7).
4. Fill the GHASH field of AES_GHASHRx with the state of the GHASH field stored at the end of the message processing.
5. Fill the IDATA field of AES_IDATARx according to the SMOD configuration used with '$\text{len}(AAD)_{64} \,||\, \text{len}(C)_{64}$' value as described in the NIST documentation and wait for DATRDY to be set; use interrupt if needed.
6. Read the GHASH field of AES_GHASHRx to obtain the current value of the hash.

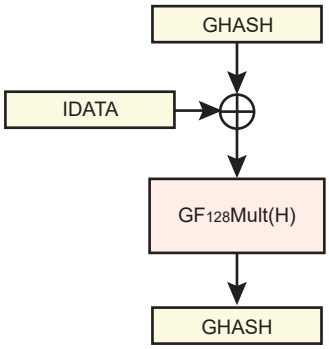
Processing $T = \text{GCTR}_K(J_0, S)$:

7. In AES_MR set OPMOD to CTR (configuration as usual for the rest).
8. Set the IV field in AES_IVRx with 'J_0' value.
9. Fill the IDATA field of AES_IDATARx with the GHASH value read at step 6 and wait for DATRDY to be set (use interrupt if needed).
10. Read the ODATA field of AES_ODATARx to obtain the GCM Tag value.

Note: Step 4 is optional if the GHASH field is to be filled with value '0' (0 length packet for instance).

Processing a Message with only AAD (GHASHH)

Figure 57-7. Single GHASHH Block Diagram (AADLEN ≤ 0x10 and CLEN = 0)



It is possible to process a message with only *AAD* setting the CLEN field to '0' in the AES_CLENR, this can be used for J_0 generation when len(IV) ≠ 96 for instance.

Example: Processing J_0 when len(IV) ≠ 96

To process J_0 = GHASH$_H$(IV || 0^{s+64} || [len(IV)]$_{64}$), the sequence is as follows:

1. In AES_MR set OPMOD to GCM and GTAGEN to '0' (configuration as usual for the rest).
2. Set KEYW in AES_KEYWRx and wait until DATRDY bit of AES_ISR is set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in the AES_GCMHRx. See Section 57.4.6.2 "Key Writing and Automatic Hash Subkey Calculation".
3. Set AADLEN field with 'len(IV || 0^{s+64} || [len(IV)]$_{64}$)' in AES_AADLENR and CLEN field to '0' in AES_CLENR. This will allow running a GHASHH only.
4. Fill the IDATA field of AES_IDATARx with the message to process (IV || 0^{s+64} || [len(IV)]$_{64}$) according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when a GHASHH step is over (use interrupt if needed).
5. Read the GHASH field of AES_GHASHRx to obtain the J_0 value.

Note: The GHASH value can be overwritten at any time by writing the GHASH field value of AES_GHASHRx, used to perform a GHASHH with an initial value for GHASH (write GHASH field between step 3 and step 4 in this case).

Processing a Single GF128 Multiplication

The AES can also be used to process a single multiplication in the Galois field on 128 bits (GF128) using a single GHASHH with custom H value (see Figure 57-7).

To run a GF128 multiplication (A x B), the sequence is as follows:

1. In AES_MR set OPMOD to GCM and GTAGEN to '0' (configuration as usual for the rest).
2. Set AADLEN field with 0x10 (16 bytes) in AES_AADLENR and CLEN field to '0' in AES_CLENR. This will allow running a single GHASHH.
3. Fill the H field of the AES_GCMHRx with B value.
4. Fill the IDATA field of AES_IDATARx with the A value according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when a GHASHH computation is over (use interrupt if needed).
5. Read the GHASH field of AES_GHASHRx to obtain the result.

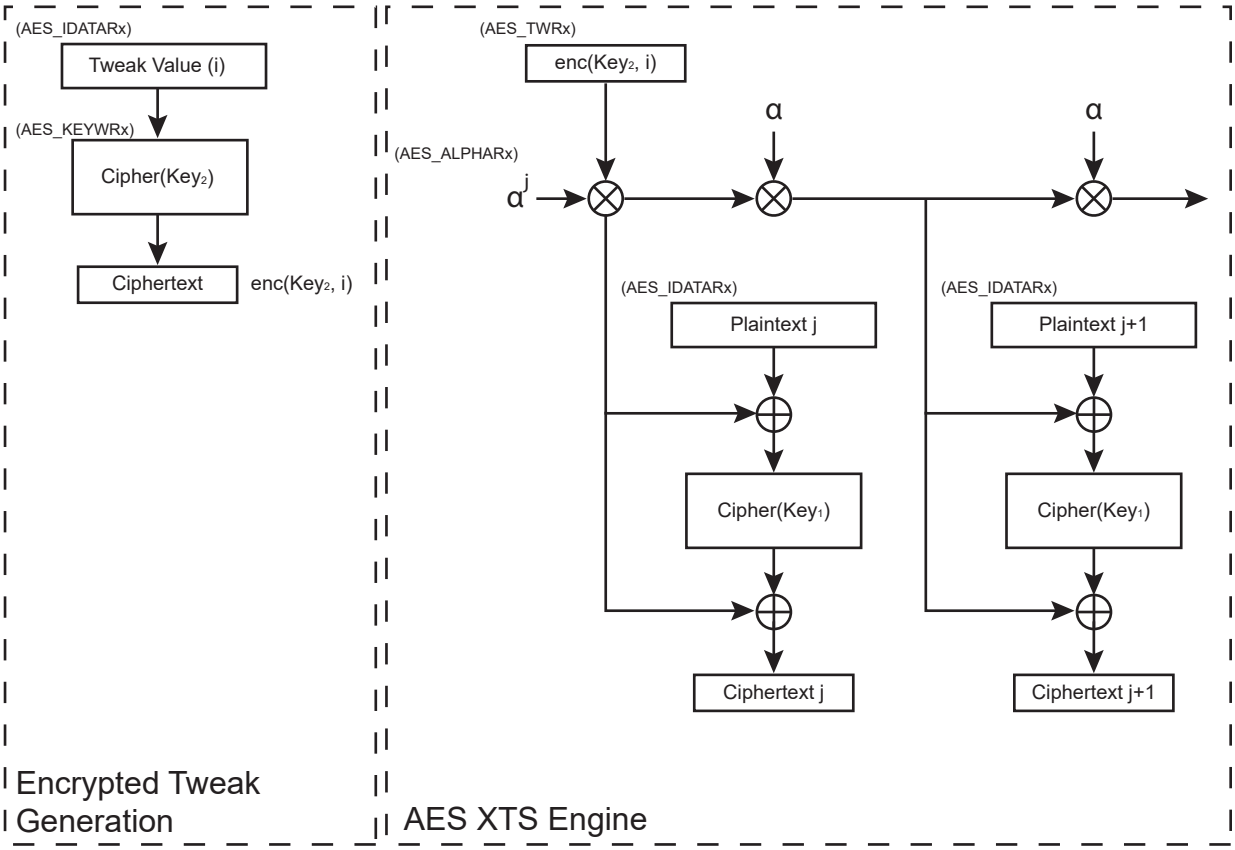
Note: The GHASH field of AES_GHASHRx can be initialized with a value C between step 3 and step 4 to run a ((A XOR C) x B) GF128 multiplication.

57.4.7 XEX-based Tweaked-codebook Mode (XTS)

XTS mode comprises the AES engine with XOR on inputs and outputs. After each encryption/decryption, the value used for the XOR is multiplied by the first GF(2^{128}) alpha primitive (0x2) and then used for the next encryption/decryption. The XTS mode uses two different keys and defines a Tweak Value (i) as additional input.

XTS processing is computed on 128-bit input data fields. There is no support for unaligned data (padding must be done manually if needed). The AES key length can be any length supported by the AES module.

Figure 57-8. XTS Block Diagram



57.4.7.1 XTS Processing Procedure

XTS processing comprises two phases:

1. Generate encrypted tweak with Key2 (this step is only required for the first processing, further consecutive processing does not require this step).
2. Process the data giving encrypted tweak and first alpha primitive for the first encryption/decryption.

Encrypted Tweak Generation

In the case of a new encryption/decryption, it is necessary to first encrypt the Tweak Value (i) with Key2. Here are the steps to follow to perform this step:

1. In AES_MR, set OPMODE to ECB, CIPHER bit to ‘1’ and other fields to the required value.
2. Set KEYW in AES_KEYWRx with Key2.
3. Fill the IDATA field of AES_IDATARx with the Tweak value (i) according to the SMOD configuration used. If Manual mode or Auto mode is used, the DATRDY bit indicates when the data have been processed and can be read in AES_ODATARx.

Data Processing

To process data using XTS mode, follow the steps below:

1. In AES_MR, set OPMODE to XTS and other fields to the required value.
2. Set KEYW in AES_KEYWRx with Key1.
3. If the data to process is the first to be processed in the data unit, or if the data block to process is not consecutive to the previous processed data block in the same data unit, then it is required to fill the AES_TWRx register with the encrypted Tweak Value (see "Encrypted Tweak Generation" for details), and then to fill the AES_ALPHARx register with the alpha primitive corresponding to the block number in the data unit. Otherwise, ignore Step 3.
4. Fill the IDATA field of AES_IDATARx with the data to process according to the SMOD configuration used. If Manual mode or Auto mode is used, the DATRDY bit indicates when the data have been processed and can be read in AES ODATARx. Repeat Step 4 as long as consecutive data blocks are processed in the same data unit.

57.4.8 Automatic Padding Mode

When Automatic Padding mode is configured, the message is automatically padded after the last block is written. Depending on the size of the message, either a padding is performed after the last part of the message and padding blocks are added or only padding blocks are added.

Both IPSEC and SSL padding standards are supported.

The auto padding feature only supports CBC and CTR mode.

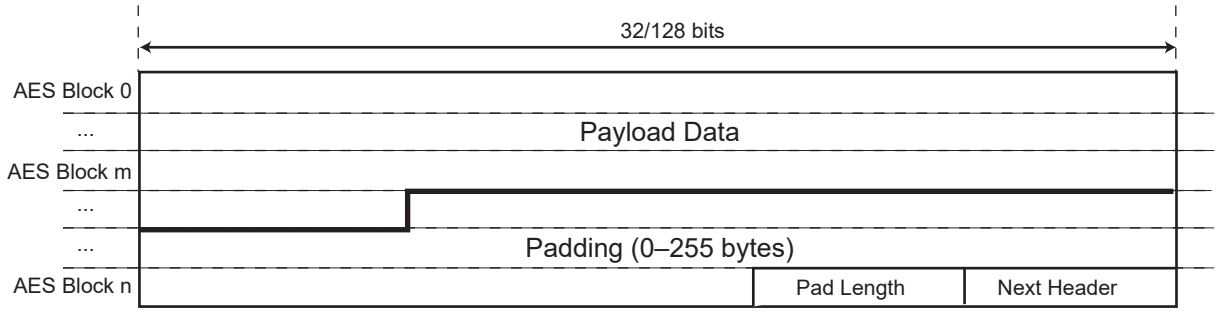
Note: When automatic padding is enabled and the field SMOD=2 in AES_MR, the bit DUALBUFF must be cleared in AES_MR.

57.4.8.1 IPSEC Padding

Automatic Padding is enabled by writing a '1' to the APEN bit in the AES Extended Mode register (AES_EMR).

IPSEC padding mode is selected by writing a '0' to the APM bit in the AES_EMR.

Figure 57-9. IPSEC Padding



The "Pad Length" in bytes can be configured in the PADLEN field and the "Next Header" value can be configured in the NHEAD field of the AES_EMR. The PADLEN field must be configured with the length of the padding section, not including the length of the "Pad Length" and "Next Header" sections.

The BCNT field in the AES Byte Counter register (AES_BCNT) defines the length, in bytes, of the message to process. It must be configured before writing the first data in AES_IDATARx and the remaining bytes to process can be read at anytime (BCNT value is decremented after each AES_IDATARx access).

AES_BCNT.BCNT and AES_EMR.PADLEN must be configured so that the sum of the length of the message (Payload Data) and of the length of the Padding, Pad Length (1 byte) and Next Header (1 byte) sections is a multiple of the AES block size (128 bits).

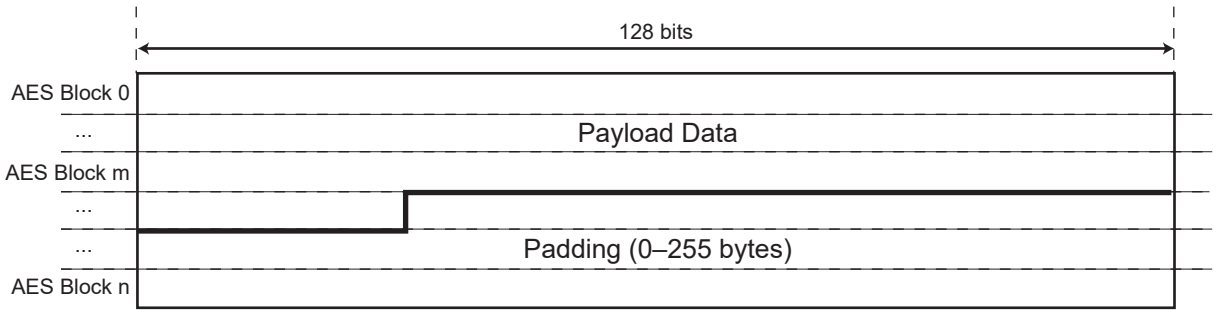
To process an IPSEC message using auto-padding, the sequence is as follows:

1. In AES_MR set OPMOD to either CBC or CTR mode.
2. In AES_EMR set APEN to ‘1’, APM to ‘0’, PADLEN to the desired padding length in byte and NHEAD to the desired Next Header field value.
3. In AES_BCNT set the BCNT field with the whole message length, without padding, in byte.
4. Set KEYW in AES_KEYWRx registers
5. Set IV in AES_IVRx registers if needed
6. Fill IDATA field of AES_IDATARx with the message to process according to the SMOD configuration used. On the last data block write only what is necessary (e.g., write only AES_IDATAR0 if last block size is ≤ 32 bits).
7. Wait for the DATRDY flag to be raised, meaning auto-padding completion and last block processing.

57.4.8.2 SSL Padding

Auto Padding is enabled by writing a ‘1’ to the APEN bit in the AES_EMR and SSL padding mode is selected by writing a ‘1’ to the APM bit in the AES_EMR.

Figure 57-10. SSL Padding



The padding length is configured in the field PADLEN of the AES_EMR.

The BCNT field in the AES Byte Counter register (AES_BCNT) defines the length, in bytes, of the message to process. It must be configured before writing the first data in AES_IDATARx and the remaining bytes to process can be read at anytime (BCNT value is decremented after each AES_IDATARx access).

AES_BCNT.BCNT and AES_EMR.PADLEN must be configured so that the length of the message plus the length of the padding section is a multiple of the AES block size (128 bits).

To process a complete SSL message, the sequence is as follows:

1. In AES_MR set OPMOD to either CBC or CTR mode.
2. In AES_EMR set APEN to ‘1’, APM to ‘1’, PADLEN to the desired padding length in bytes.
3. In AES_BCNT set the BCNT field with the whole message length, without padding, in bytes.
4. Set KEYW in AES_KEYWRx
5. Set IV in AES_IVRx if needed
6. Fill IDATA field of AES_IDATARx with the message to process according to the SMOD configuration used. On the last data block write only what is necessary (e.g., write only AES_IDATAR0 if last block size is ≤ 32 bits).
7. Wait for the DATRDY flag to be raised, meaning auto-padding completion and last block processing.

57.4.8.3 Flags

The EOPAD flag in the AES_ISR rises as soon as the automatic padding phase is over, meaning that all the extra padding blocks have been processed. Reading the AES_ISR clears this flag.

The PLENERR flag in the AES_ISR indicates an error in the frame configuration, meaning that the whole message length including padding does not respect the standard selected. The PLENERR flag rises at the end of the frame in case of wrong message length and is cleared reading the AES_ISR.

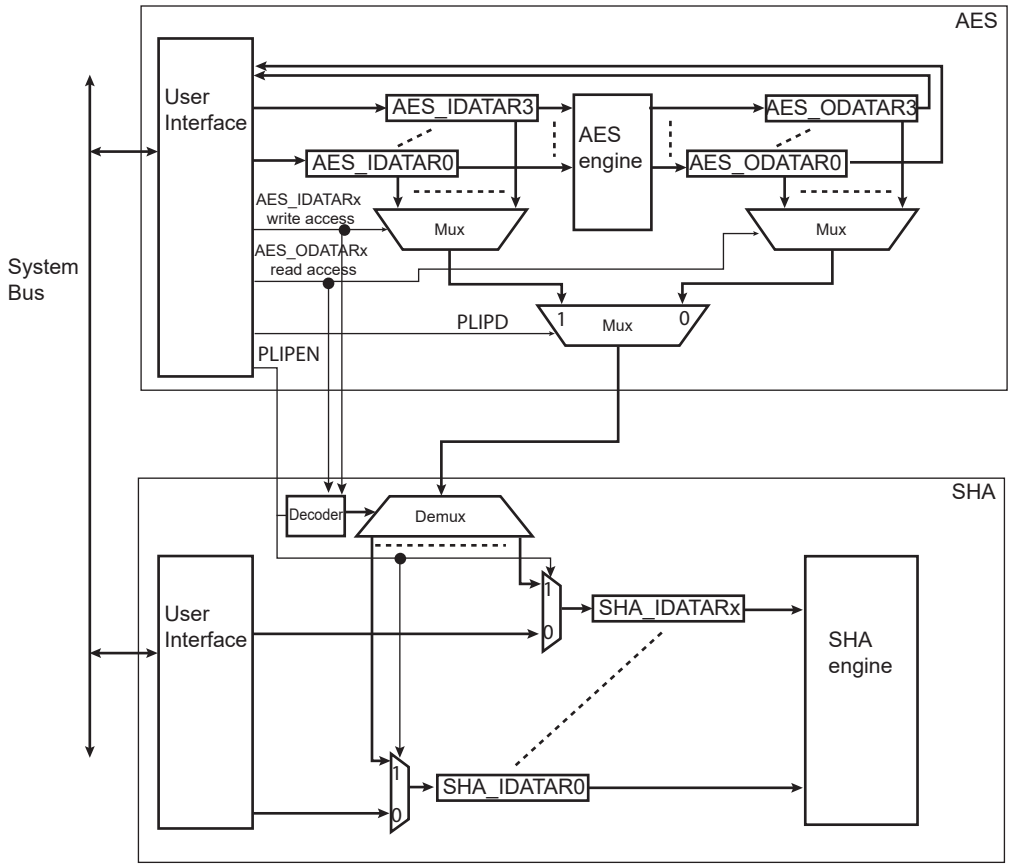
In IPSEC/SSL standard message length including padding must be a multiple of the AES block size when CBC mode is used and multiple of 32-bit if CTR mode is used.

57.4.9 Secure Protocol Layers Improved Performances

Secure protocol layers such as IPSec require encryption and authentication. For IPSec, the authentication is based on HMAC, thus SHA is required. To optimize performance, the AES embeds a mode of operation that enables the SHA module to process the input or output data of the AES module. If this mode is enabled, write access is required only into the AES_IDATARx registers, since SHA input data registers are automatically written by AES without software intervention. When the DMA is configured to transfer a buffer of data (input frame), only one transfer descriptor is required for both authentication and encryption/decryption processes and only one buffer is transferred through the system bus (reducing the load of the system bus).

Improved performance for secure protocol layers requires the bit PLIPEN to be set in AES_EMR.

Figure 57-11. Secure Protocol Layers Improved Performances Block Diagram



57.4.9.1 Cipher Mode

When bit PLIPD is cleared and bit PLIPEN=1, the message written into AES_IDATARx is first encrypted with the AES module and the encrypted message is authenticated with the SHA module. Therefore, when PLIPD is cleared, the AES_ODATARx are selected and sent to SHA_IDATARx as soon as AES_ODATARx are read. A read access in AES corresponds to a write access to the corresponding SHA_IDATARx. The number of SHA_IDATARx is greater than the number of AES_ODATARx, but the SHA module embeds the decoding logic to automatically dispatch the AES_ODATARx values into the corresponding SHA_IDATARx without software intervention.

57.4.9.2 Decipher Mode

When bit PLIPD is written to one and bit PLIPEN=1, the message written into AES_IDATARx is decrypted with the AES module and also sent to SHA for authentication. Therefore, when PLIPD=1, the AES_IDATARx are selected and sent to SHA_IDATARx as soon as AES_IDATARx are written. A write access in AES corresponds to a write access to the corresponding SHA_IDATARx. The number of SHA_IDATARx is greater than the number of AES_ODATARx, but the SHA module embeds the decoding logic to automatically dispatch the AES_IDATARx values into the corresponding SHA_IDATARx without software intervention.

57.4.9.3 Encapsulating Security Payload (ESP) IPSec Examples

The following examples describe how to configure AES and SHA to optimize processing an ESP IPSec frame for maximum performance.

The cipher (or decipher) of an ESP IPSec frame requires both encryption (or decryption) and authentication.

For cipher, the input frame located in the system memory must first be padded and the resulting buffer encrypted. The encrypted frame must be written back to the system memory and sent to the authentication module.

When the AES module is configured to improve the performance of the secure protocol layers (bit PLIPEN = 1), the data transfers are simplified, limiting the bandwidth requirements on the system bus.

Before configuring the DMA to start the transfer of the data buffer (input frame) to the AES, the following actions must be taken in registers:

- The BCNT field in AES_BCNT must be configured with the length of the message (Input Frame).
- The padding length of the AES must be configured in field PADLEN of AES_EMR. See Section 57.4.8 “Automatic Padding Mode” to configure Automatic padding mode.
- The next header value must be configured in field NHEAD of AES_EMR.
- Field SMOD in AES_MR and field SMOD in SHA_MR must be configured to 2.

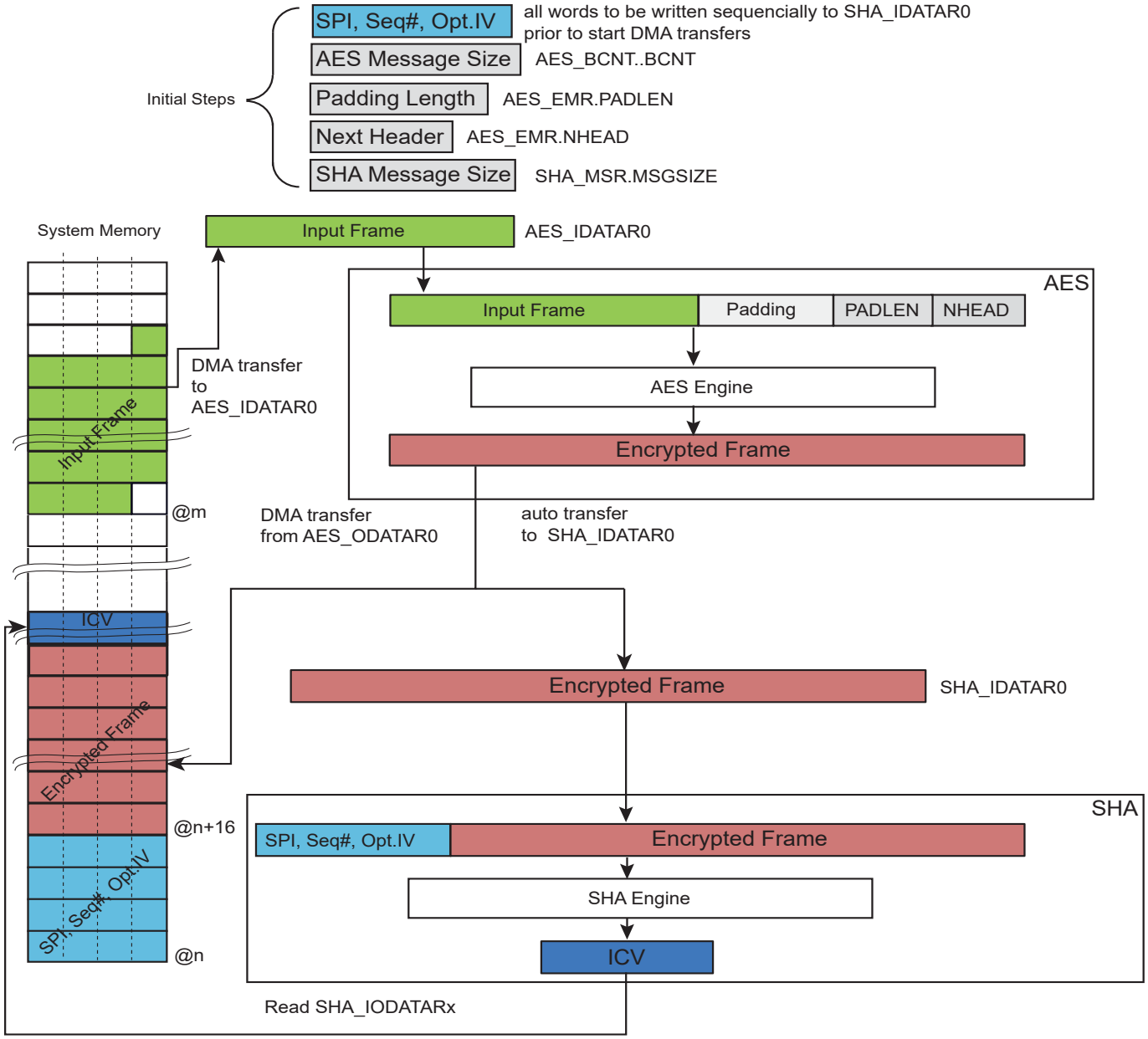
Note: When automatic padding is enabled and the field SMOD = 2 in AES_MR, the bit DUALBUFF must be cleared in AES_MR.

- The MSGSIZE field in SHA_MSR must be configured with the length of the authentication message including the optional extended sequence number (ESN) and header and trailer information required by the authentication algorithm used (HMAC, etc.). Refer to the section “Secure Hash Algorithm (SHA)” in this datasheet for more details on configurations for optimized processing of the header information.
- The Security Parameter Index (SPI, sequence number (SEQ#)) and the optional Initialization Vector (IV) must be configured sequentially in SHA_IDATAR0.
- A first DMA transfer descriptor must be configured to transfer the input frame from the system memory to the AES input data registers (AES_IDATARx), and a second DMA descriptor must be configured to transfer the encrypted frame from AES to the system memory.

Note: If the bit PLIPEN = 1 in AES_EMR, there is no need to define a transfer descriptor to load the encrypted frame into the SHA input data registers because the transfer is automatically performed while the second descriptor transfer is in progress.

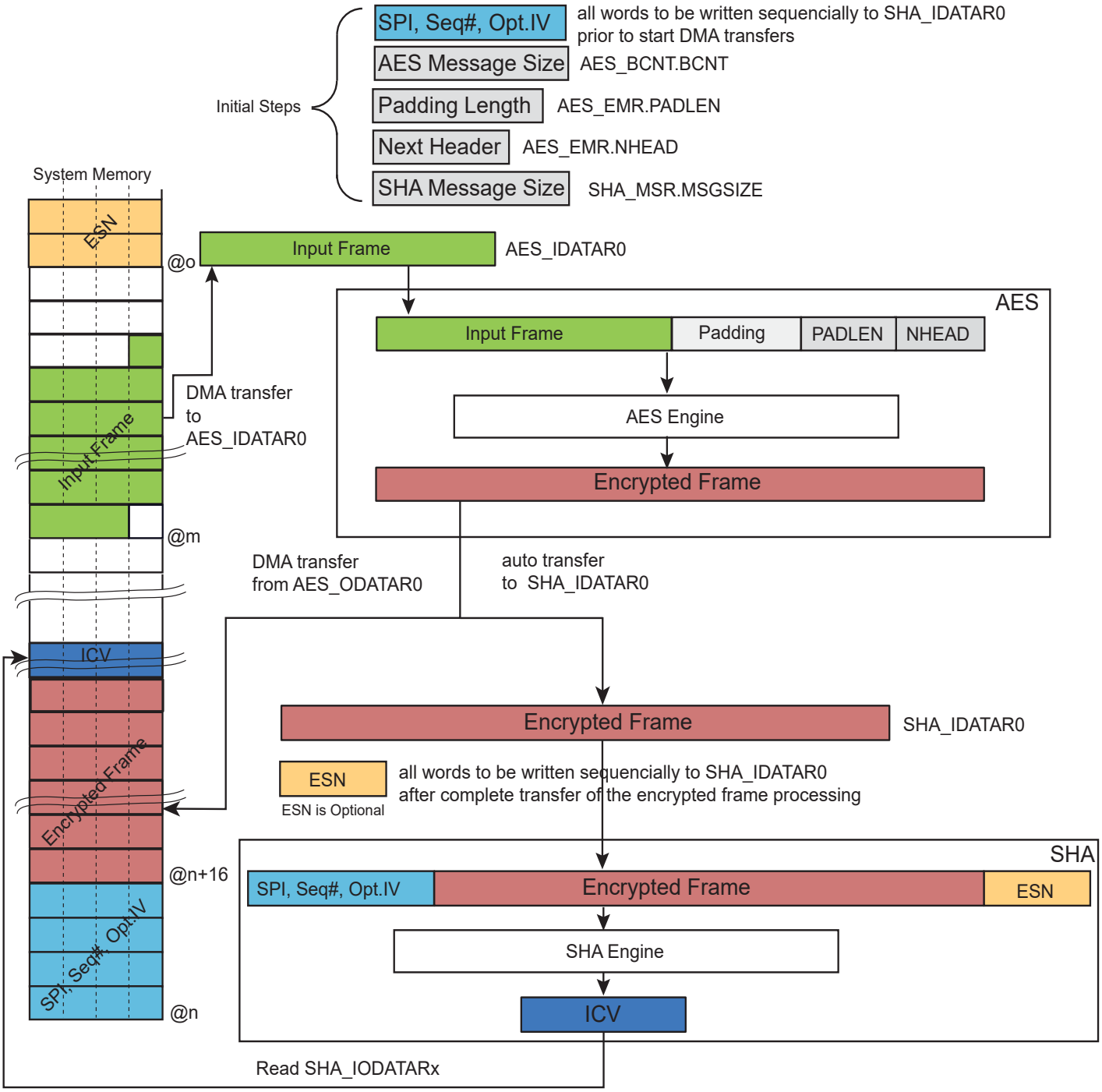
See Figure 57-12 and Figure 57-13.

Figure 57-12. Generation of an ESP IPSec Frame without ESN



If the optional extended sequence number is required for authentication, wait for the AES-to-system memory DMA buffer transfer to complete before configuring the ESN value. The ESN value must be configured in the SHA by writing sequentially each 32-bit word of the ESN into the SHA_IDATAR0 register. Wait for flag WRDY=1 in SHA_ISR before each write in the SHA_IDATAR0 register. Refer to Figure 57-13.

Figure 57-13. Generation of an ESP IPSec Frame with ESN



To decipher an ESP IPSec frame without the optional ESN trailer information, two DMA channels are required and the SHA must be configured in Automatic padding mode.

Note: AES automatic padding must be disabled when deciphering a frame.

- A first DMA transfer descriptor must be configured to load the received encrypted frame from the system memory to AES_IDATARx for decryption. The start address of the first transfer descriptor must be defined after the SPI, SEQ#, and optional IV (see Figure 57-14).
- A second DMA descriptor must be configured to transfer the decrypted frame from AES_ODATARx to the system memory.
- PLIPEN and PLIPD must be written to ‘1’ so that the data buffer is written in AES_IDATARx and in SHA_IDATARx.

The SHA has the capability to perform an automatic check with an expected integrity check value if this value is appended at the end of the frame buffer (SHA_MR.CHECK=2). Thus, if the first transfer descriptor includes the ICV for SHA, the first DMA transfer allows the decryption and authentication processes including the automatic check. The decrypted part resulting from ICV is not required for downstream processing and must be considered as dummy data.

The end of the decryption and authentication processes occur when flag CHECKF=1 in SHA_ISR. The authentication status is provided by field CHKST in SHA_ISR.

If the optional ESN trailer information is part of the ICV (see Figure 57-15), the ESN must be manually written into SHA_IDATAR0. The ESN value must be written after completion of the system memory-to-AES DMA buffer transfer. The ESN value must be configured in the SHA by writing sequentially each 32-bit word of the ESN into the SHA_IDATAR0 register. Wait for flag WRDY=1 in SHA_ISR before each write in the SHA_IDATAR0 register.

When the optional ESN trailer information is part of the ICV, it is not possible to include the ICV received in the input frame to the first transfer descriptor. Moreover, if the HMAC algorithm is used for authentication, no automatic check can be performed when optimizing the processing performances of the SHA module. For more details, see the section “Secure Hash Algorithm (SHA)” in this datasheet. The result of the HMAC read in the SHA_IODATARx must be manually compared with the ICV value of the input frame. The comparison must be performed after the end of the authentication process. The authentication process is completed when the DATRDY flag is set in SHA_ISR.

Figure 57-14. Decryption of an ESP IP Sec Frame without ESN

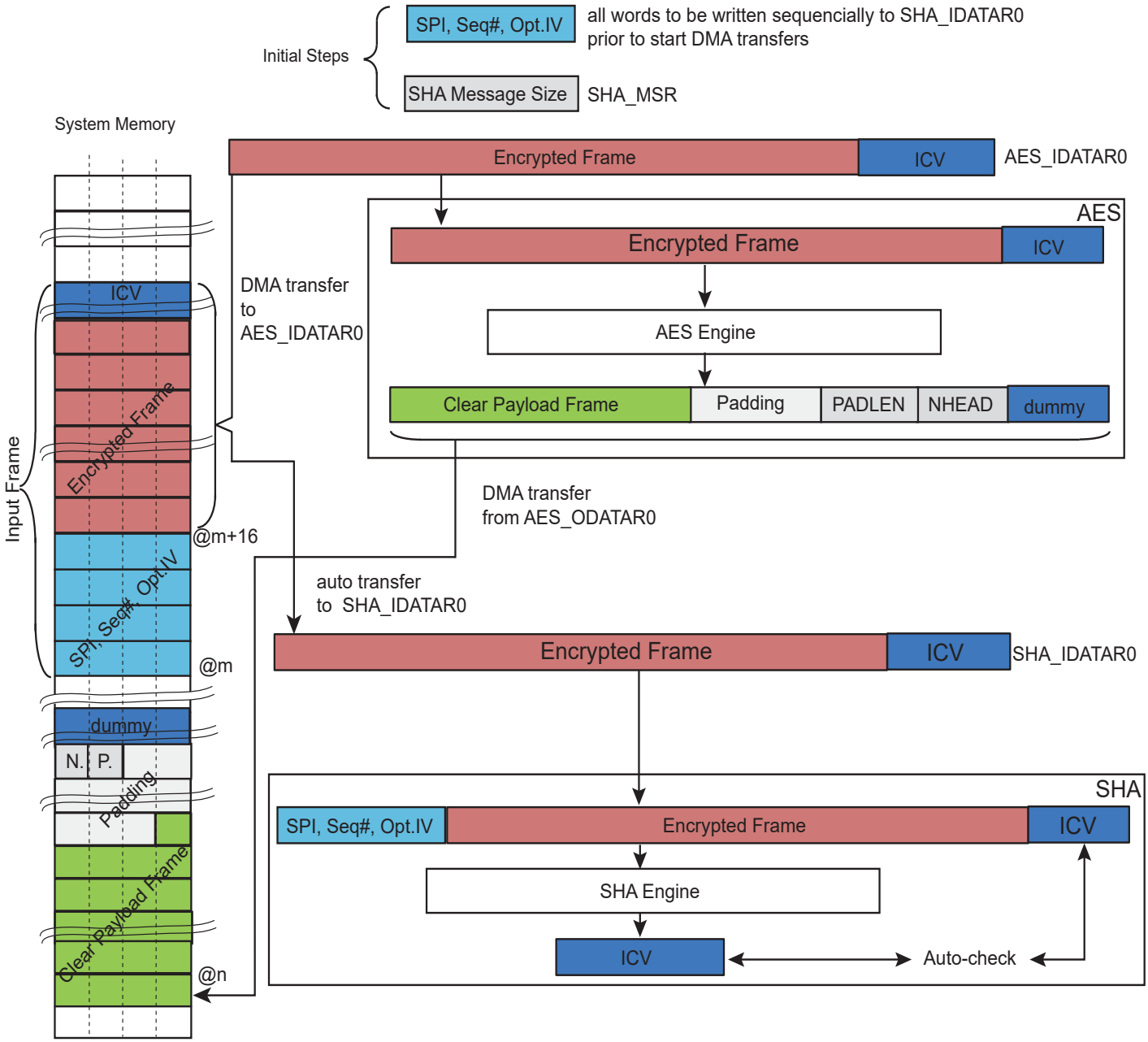
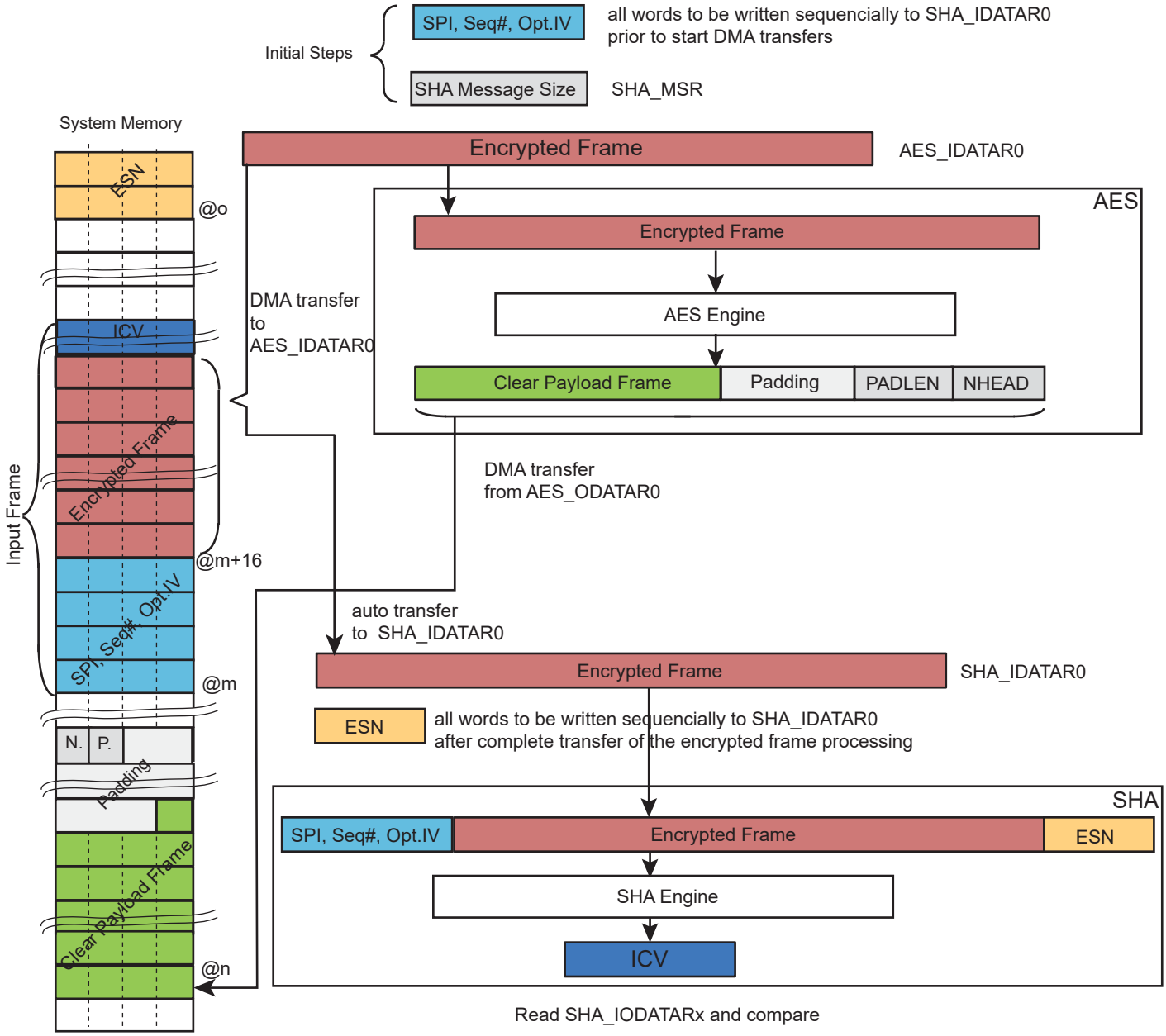


Figure 57-15. Decryption of an ESP IPSec Frame with ESN



57.4.10 Security Features

57.4.10.1 Unspecified Register Access Detection

When an unspecified register access occurs, the URAD flag in the AES_ISR is raised. Its source is then reported in the Unspecified Register Access Type (URAT) field. Only the last unspecified register access is available through the URAT field.

Several kinds of unspecified register accesses can occur:

- Input Data register written during the data processing when SMOD = IDATAR0_START
- Output Data register read during data processing
- Mode register written during data processing
- Output Data register read during subkeys generation
- Mode register written during subkeys generation
- Write-only register read access

The URAD bit and the URAT field can only be reset by the SWRST bit in the AES_CR.

57.5 Advanced Encryption Standard (AES) User Interface

Table 57-5. Register Mapping

Offset	Register	Name	Access	Reset
0x00	Control Register	AES_CR	Write-only	–
0x04	Mode Register	AES_MR	Read/Write	0x0
0x08–0x0C	Reserved	–	–	–
0x10	Interrupt Enable Register	AES_IER	Write-only	–
0x14	Interrupt Disable Register	AES_IDR	Write-only	–
0x18	Interrupt Mask Register	AES_IMR	Read-only	0x0
0x1C	Interrupt Status Register	AES_ISR	Read-only	0x0
0x20	Key Word Register 0	AES_KEYWR0	Write-only	–
0x24	Key Word Register 1	AES_KEYWR1	Write-only	–
0x28	Key Word Register 2	AES_KEYWR2	Write-only	–
0x2C	Key Word Register 3	AES_KEYWR3	Write-only	–
0x30	Key Word Register 4	AES_KEYWR4	Write-only	–
0x34	Key Word Register 5	AES_KEYWR5	Write-only	–
0x38	Key Word Register 6	AES_KEYWR6	Write-only	–
0x3C	Key Word Register 7	AES_KEYWR7	Write-only	–
0x40	Input Data Register 0	AES_IDATAR0	Write-only	–
0x44	Input Data Register 1	AES_IDATAR1	Write-only	–
0x48	Input Data Register 2	AES_IDATAR2	Write-only	–
0x4C	Input Data Register 3	AES_IDATAR3	Write-only	–
0x50	Output Data Register 0	AES_ODATAR0	Read-only	0x0
0x54	Output Data Register 1	AES_ODATAR1	Read-only	0x0
0x58	Output Data Register 2	AES_ODATAR2	Read-only	0x0
0x5C	Output Data Register 3	AES_ODATAR3	Read-only	0x0
0x60	Initialization Vector Register 0	AES_IVR0	Write-only	–
0x64	Initialization Vector Register 1	AES_IVR1	Write-only	–
0x68	Initialization Vector Register 2	AES_IVR2	Write-only	–
0x6C	Initialization Vector Register 3	AES_IVR3	Write-only	–
0x70	Additional Authenticated Data Length Register	AES_AADLENR	Read/Write	–
0x74	Plaintext/Ciphertext Length Register	AES_CLENR	Read/Write	–
0x78	GCM Intermediate Hash Word Register 0	AES_GHASHR0	Read/Write	–
0x7C	GCM Intermediate Hash Word Register 1	AES_GHASHR1	Read/Write	–
0x80	GCM Intermediate Hash Word Register 2	AES_GHASHR2	Read/Write	–
0x84	GCM Intermediate Hash Word Register 3	AES_GHASHR3	Read/Write	–
0x88	GCM Authentication Tag Word Register 0	AES_TAGR0	Read-only	–
0x8C	GCM Authentication Tag Word Register 1	AES_TAGR1	Read-only	–

Table 57-5. Register Mapping (Continued)

Offset	Register	Name	Access	Reset
0x90	GCM Authentication Tag Word Register 2	AES_TAGR2	Read-only	–
0x94	GCM Authentication Tag Word Register 3	AES_TAGR3	Read-only	–
0x98	GCM Encryption Counter Value Register	AES_CTRR	Read-only	–
0x9C	GCM H Word Register 0	AES_GCMHR0	Read/Write	–
0xA0	GCM H Word Register 1	AES_GCMHR1	Read/Write	–
0xA4	GCM H Word Register 2	AES_GCMHR2	Read/Write	–
0xA8	GCM H Word Register 3	AES_GCMHR3	Read/Write	–
0xAC	Reserved	–	–	–
0xB0	Extended Mode Register	AES_EMR	Read/Write	0x0
0xB4	Byte Counter Register	AES_BCNT	Read/Write	0x0
0xC0	Tweak Word Register 0	AES_TWR0	Read/Write	0x0
0xC4	Tweak Word Register 1	AES_TWR1	Read/Write	0x0
0xC8	Tweak Word Register 2	AES_TWR2	Read/Write	0x0
0xCC	Tweak Word Register 3	AES_TWR3	Read/Write	0x0
0xD0	Alpha Word Register 0	AES_ALPHAR0	Write-only	–
0xD4	Alpha Word Register 1	AES_ALPHAR1	Write-only	–
0xD8	Alpha Word Register 2	AES_ALPHAR2	Write-only	–
0xDC	Alpha Word Register 3	AES_ALPHAR3	Write-only	–
0xE0	Reserved	–	–	–
0xE4–0xF8	Reserved	–	–	–
0xFC	Reserved	–	–	–

57.5.1 AES Control Register

Name: AES_CR

Address: 0xF002C000

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	SWRST
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	START

- **START: Start Processing**

0: No effect.

1: Starts manual encryption/decryption process.

- **SWRST: Software Reset**

0: No effect.

1: Resets the AES. A software-triggered hardware reset of the AES interface is performed.

57.5.2 AES Mode Register

Name: AES_MR

Address: 0xF002C004

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
CKEY				–	CFBS		
15	14	13	12	11	10	9	8
LOD	OPMOD			KEYSIZE		SMOD	
7	6	5	4	3	2	1	0
PROCDLY				DUALBUFF	–	GTAGEN	CIPHER

- **CIPHER: Processing Mode**

0: Decrypts data.

1: Encrypts data.

- **GTAGEN: GCM Automatic Tag Generation Enable**

0: Automatic GCM Tag generation disabled.

1: Automatic GCM Tag generation enabled.

- **DUALBUFF: Dual Input Buffer**

Value	Name	Description
0	INACTIVE	AES_IDATARx cannot be written during processing of previous block.
1	ACTIVE	AES_IDATARx can be written during processing of previous block when SMOD = 2. It speeds up the overall runtime of large files.

- **PROCDLY: Processing Delay**

Processing Time = N × (PROCDLY + 1)

where

N = 10 when KEYSIZE = 0

N = 12 when KEYSIZE = 1

N = 14 when KEYSIZE = 2

The processing time represents the number of clock cycles that the AES needs in order to perform one encryption/decryption.

Note: The best performance is achieved with PROCDLY equal to 0.

- **SMOD: Start Mode**

Value	Name	Description
0	MANUAL_START	Manual Mode
1	AUTO_START	Auto Mode
2	IDATAR0_START	AES_IDATAR0 access only Auto Mode (DMA)

Values which are not listed in the table must be considered as “reserved”.

If a DMA transfer is used, configure SMOD to 0x2. Refer to Section 57.4.4.3 “DMA Mode” for more details.

- **KEYSIZE: Key Size**

Value	Name	Description
0	AES128	AES Key Size is 128 bits
1	AES192	AES Key Size is 192 bits
2	AES256	AES Key Size is 256 bits

Values which are not listed in the table must be considered as “reserved”.

- **OPMOD: Operating Mode**

Value	Name	Description
0	ECB	ECB: Electronic Code Book mode
1	CBC	CBC: Cipher Block Chaining mode
2	OFB	OFB: Output Feedback mode
3	CFB	CFB: Cipher Feedback mode
4	CTR	CTR: Counter mode (16-bit internal counter)
5	GCM	GCM: Galois/Counter mode
6	XTS	XTS: XEX-based tweaked-codebook mode

Values which are not listed in the table must be considered as “reserved”.

For CBC-MAC operating mode, set OPMOD to CBC and LOD to 1.

- **LOD: Last Output Data Mode**

0: No effect.

After each end of encryption/decryption, the output data are available either on the output data registers (Manual and Auto modes) or at the address specified in the Channel Buffer Transfer Descriptor for DMA mode.

In Manual and Auto modes, the DATRDY flag is cleared when at least one of the Output Data registers is read.

1: The DATRDY flag is cleared when at least one of the Input Data Registers is written.

No more Output Data Register reads is necessary between consecutive encryptions/decryptions (see Section 57.4.5 “Last Output Data Mode”).

Warning: In DMA mode, reading to the Output Data registers before the last data encryption/decryption process may lead to unpredictable results.

- **CFBS: Cipher Feedback Data Size**

Value	Name	Description
0	SIZE_128BIT	128-bit
1	SIZE_64BIT	64-bit
2	SIZE_32BIT	32-bit
3	SIZE_16BIT	16-bit
4	SIZE_8BIT	8-bit

Values which are not listed in the table must be considered as “reserved”.

- **CKEY: Key**

Value	Name	Description
0xE	PASSWD	This field must be written with 0xE the first time the AES_MR is programmed. For subsequent programming of the AES_MR, any value can be written, including that of 0xE. Always reads as 0.

57.5.3 AES Interrupt Enable Register

Name: AES_IER

Address: 0xF002C010

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	PLENERR	EOPAD	TAGRDY
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	URAD
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

- **DATRDY: Data Ready Interrupt Enable**
- **URAD: Unspecified Register Access Detection Interrupt Enable**
- **TAGRDY: GCM Tag Ready Interrupt Enable**
- **EOPAD: End of Padding Interrupt Enable**
- **PLENERR: Padding Length Error Interrupt Enable**

57.5.4 AES Interrupt Disable Register

Name: AES_IDR

Address: 0xF002C014

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	PLENERR	EOPAD	TAGRDY
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	URAD
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

- **DATRDY: Data Ready Interrupt Disable**
- **URAD: Unspecified Register Access Detection Interrupt Disable**
- **TAGRDY: GCM Tag Ready Interrupt Disable**
- **EOPAD: End of Padding Interrupt Disable**
- **PLENERR: Padding Length Error Interrupt Disable**

57.5.5 AES Interrupt Mask Register

Name: AES_IMR

Address: 0xF002C018

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	PLENERR	EOPAD	TAGRDY
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	URAD
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

- **DATRDY: Data Ready Interrupt Mask**
- **URAD: Unspecified Register Access Detection Interrupt Mask**
- **TAGRDY: GCM Tag Ready Interrupt Mask**
- **EOPAD: End of Padding Interrupt Mask**
- **PLENERR: Padding Length Error Interrupt Mask**

57.5.6 AES Interrupt Status Register

Name: AES_ISR

Address: 0xF002C01C

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	PLENERR	EOPAD	TAGRDY
15	14	13	12	11	10	9	8
URAT				–	–	–	URAD
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

- **DATRDY: Data Ready (cleared by setting bit START or bit SWRST in AES_CR or by reading AES_ODATARx)**

0: Output data not valid.

1: Encryption or decryption process is completed.

Note: If AES_MR.LOD = 1: In Manual and Auto mode, the DATRDY flag can also be cleared by writing at least one AES_IDATARx.

- **URAD: Unspecified Register Access Detection Status (cleared by writing SWRST in AES_CR)**

0: No unspecified register access has been detected since the last SWRST.

1: At least one unspecified register access has been detected since the last SWRST.

- **URAT: Unspecified Register Access (cleared by writing SWRST in AES_CR)**

Value	Name	Description
0	IDR_WR_PROCESSING	Input Data register written during the data processing when SMOD = 0x2 mode.
1	ODR_RD_PROCESSING	Output Data register read during the data processing.
2	MR_WR_PROCESSING	Mode register written during the data processing.
3	ODR_RD_SUBKGEN	Output Data register read during the subkeys generation.
4	MR_WR_SUBKGEN	Mode register written during the subkeys generation.
5	WOR_RD_ACCESS	Write-only register read access.

Only the last Unspecified Register Access Type is available through the URAT field.

- **TAGRDY: GCM Tag Ready**

0: GCM Tag is not valid.

1: GCM Tag generation is complete (cleared by reading GCM Tag, starting another processing or when writing a new key).

- **EOPAD: End of Padding**

0: Padding is not over.

1: Padding phase is over.

- **PLENERR: Padding Length Error**

0: No Padding Length Error occurred.

1: Padding Length Error detected.

57.5.7 AES Key Word Register x

Name: AES_KEYWRx [x=0..7]

Address: 0xF002C020

Access: Write-only

31	30	29	28	27	26	25	24
KEYW							
23	22	21	20	19	18	17	16
KEYW							
15	14	13	12	11	10	9	8
KEYW							
7	6	5	4	3	2	1	0
KEYW							

- **KEYW: Key Word**

The four/six/eight 32-bit Key Word registers set the 128-bit/192-bit/256-bit cryptographic key used for AES encryption/decryption.

AES_KEYWR0 corresponds to the first word of the key and respectively AES_KEYWR3/AES_KEYWR5/AES_KEYWR7 to the last one.

Whenever a new key (AES_KEYWRx) is written to the hardware, two automatic actions are processed:

- GCM hash subkey generation
- AES_GHASHRx Clear

See Section 57.4.6.2 “Key Writing and Automatic Hash Subkey Calculation” for details.

These registers are write-only to prevent the key from being read by another application.

57.5.8 AES Input Data Register x

Name: AES_IDATARx [x=0..3]

Address: 0xF002C040

Access: Write-only

31	30	29	28	27	26	25	24
IDATA							
23	22	21	20	19	18	17	16
IDATA							
15	14	13	12	11	10	9	8
IDATA							
7	6	5	4	3	2	1	0
IDATA							

- **IDATA: Input Data Word**

The four 32-bit Input Data registers set the 128-bit data block used for encryption/decryption.

AES_IDATAR0 corresponds to the first word of the data to be encrypted/decrypted, and AES_IDATAR3 to the last one.

These registers are write-only to prevent the input data from being read by another application.

57.5.9 AES Output Data Register x

Name: AES_ODATARx [x=0..3]

Address: 0xF002C050

Access: Read-only

31	30	29	28	27	26	25	24
ODATA							
23	22	21	20	19	18	17	16
ODATA							
15	14	13	12	11	10	9	8
ODATA							
7	6	5	4	3	2	1	0
ODATA							

- **ODATA: Output Data**

The four 32-bit Output Data registers contain the 128-bit data block that has been encrypted/decrypted.

AES_ODATAR0 corresponds to the first word, AES_ODATAR3 to the last one.

57.5.10 AES Initialization Vector Register x

Name: AES_IVRx [x=0..3]

Address: 0xF002C060

Access: Write-only

31	30	29	28	27	26	25	24
IV							
23	22	21	20	19	18	17	16
IV							
15	14	13	12	11	10	9	8
IV							
7	6	5	4	3	2	1	0
IV							

- **IV: Initialization Vector**

The four 32-bit Initialization Vector registers set the 128-bit Initialization Vector data block that is used by some modes of operation as an additional initial input.

AES_IVR0 corresponds to the first word of the Initialization Vector, AES_IVR3 to the last one.

These registers are write-only to prevent the Initialization Vector from being read by another application.

For CBC, OFB and CFB modes, the IV input value corresponds to the initialization vector.

For CTR mode, the IV input value corresponds to the initial counter value.

Note: These registers are not used in ECB mode and must not be written.

57.5.11 AES Additional Authenticated Data Length Register

Name: AES_AADLENR

Address: 0xF002C070

Access: Read/Write

31	30	29	28	27	26	25	24
AADLEN							
23	22	21	20	19	18	17	16
AADLEN							
15	14	13	12	11	10	9	8
AADLEN							
7	6	5	4	3	2	1	0
AADLEN							

- **AADLEN: Additional Authenticated Data Length**

Length in bytes of the Additional Authenticated Data (*AAD*) that is to be processed.

Note: The maximum byte length of the *AAD* portion of a message is limited to the 32-bit counter length.

57.5.12 AES Plaintext/Ciphertext Length Register

Name: AES_CLENR

Address: 0xF002C074

Access: Read/Write

31	30	29	28	27	26	25	24
CLEN							
23	22	21	20	19	18	17	16
CLEN							
15	14	13	12	11	10	9	8
CLEN							
7	6	5	4	3	2	1	0
CLEN							

- **CLEN: Plaintext/Ciphertext Length**

Length in bytes of the plaintext/ciphertext (*C*) data that is to be processed.

Note: The maximum byte length of the C portion of a message is limited to the 32-bit counter length.

57.5.13 AES GCM Intermediate Hash Word Register x

Name: AES_GHASHRx [x=0..3]

Address: 0xF002C078

Access: Read/Write

31	30	29	28	27	26	25	24
GHASH							
23	22	21	20	19	18	17	16
GHASH							
15	14	13	12	11	10	9	8
GHASH							
7	6	5	4	3	2	1	0
GHASH							

- **GHASH: Intermediate GCM Hash Word x**

The four 32-bit Intermediate Hash Word registers expose the intermediate GHASH value. May be read to save the current GHASH value so processing can later be resumed, presumably on a later message fragment. Whenever a new key (AES_KEYWRx) is written to the hardware two automatic actions are processed:

- GCM hash subkey generation
- AES_GHASHRx Clear

See Section 57.4.6.2 “Key Writing and Automatic Hash Subkey Calculation” for details.

If an application software-specific hash initial value is needed for the GHASH, it must be written to the AES_GHASHRx:

- after a write to AES_KEYWRx, if any,
- before starting the input data feed.

57.5.14 AES GCM Authentication Tag Word Register x

Name: AES_TAGRx [x=0..3]

Address: 0xF002C088

Access: Read-only

31	30	29	28	27	26	25	24
TAG							

23	22	21	20	19	18	17	16
TAG							

15	14	13	12	11	10	9	8
TAG							

7	6	5	4	3	2	1	0
TAG							

- **TAG: GCM Authentication Tag x**

The four 32-bit Tag registers contain the final 128-bit GCM Authentication tag (*T*) when GCM processing is complete. TAG0 corresponds to the first word, TAG3 to the last word.

57.5.15 AES GCM Encryption Counter Value Register

Name: AES_CTRR

Address: 0xF002C098

Access: Read-only

31	30	29	28	27	26	25	24
CTR							
23	22	21	20	19	18	17	16
CTR							
15	14	13	12	11	10	9	8
CTR							
7	6	5	4	3	2	1	0
CTR							

- **CTR: GCM Encryption Counter**

Reports the current value of the 32-bit GCM counter.

57.5.16 AES GCM H Word Register x

Name: AES_GCMHRx [x=0..3]

Address: 0xF002C09C

Access: Read/Write

31	30	29	28	27	26	25	24
H							
23	22	21	20	19	18	17	16
H							
15	14	13	12	11	10	9	8
H							
7	6	5	4	3	2	1	0
H							

- **H: GCM H Word x**

The four 32-bit H Word registers contain the 128-bit GCM hash subkey *H* value.

Whenever a new key (AES_KEYWRx) is written to the hardware, two automatic actions are processed:

- GCM hash subkey *H* generation
- AES_GHASHRx Clear

If the application software requires a specific hash subkey, the automatically-generated H value can be overwritten in the AES_GCMHRx. See Section 57.4.6.2 “Key Writing and Automatic Hash Subkey Calculation” for details.

Generating a GCM hash subkey *H* by a write in the AES_GCMHRx enables to:

- select the GCM hash subkey *H* for GHASH operations,
- select one operand to process a single GF128 multiply.

57.5.17 AES Extended Mode Register

Name: AES_EMR

Address: 0xF002C0B0

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
NHEAD							
15	14	13	12	11	10	9	8
PADLEN							
7	6	5	4	3	2	1	0
–	–	PLIPD	PLIPEN	–	–	APM	APEN

- **APEN: Auto Padding Enable**

0: Auto Padding feature is disabled.

1: Auto Padding feature is enabled.

- **APM: Auto Padding Mode**

0: Auto Padding performed according to IPSEC standard.

1: Auto Padding performed according to SSL standard.

- **PLIPEN: Protocol Layer Improved Performance Enable**

0: Protocol layer improved performance is disabled.

1: Protocol layer improved performance is enabled.

- **PLIPD: Protocol Layer Improved Performance Decipher**

0: Protocol layer improved performance is in ciphering mode.

1: Protocol layer improved performance is in deciphering mode.

- **PADLEN: Auto Padding Length**

0–255: Padding Length in bytes

- **NHEAD: IPSEC Next Header**

0–255: IPSEC Next Header field

57.5.18 AES Byte Counter Register

Name: AES_BCNT

Address: 0xF002C0B4

Access: Read/Write

31	30	29	28	27	26	25	24
BCNT							
23	22	21	20	19	18	17	16
BCNT							
15	14	13	12	11	10	9	8
BCNT							
7	6	5	4	3	2	1	0
BCNT							

- **BCNT: Auto Padding Byte Counter**

Auto padding byte counter value. BCNT must be greater than 0.

57.5.19 AES Tweak Word Register x

Name: AES_TWRx [x=0..3]

Address: 0xF002C0C0

Access: Read/Write

31	30	29	28	27	26	25	24
TWEAK							

23	22	21	20	19	18	17	16
TWEAK							

15	14	13	12	11	10	9	8
TWEAK							

7	6	5	4	3	2	1	0
TWEAK							

- **TWEAK: Tweak Word x**

The four 32-bit Tweak Word registers contain the 128-bit Tweak value.

57.5.20 AES Alpha Word Register x

Name: AES_ALPHARx [x=0..3]

Address: 0xF002C0D0

Access: Write-only

31	30	29	28	27	26	25	24
ALPHA							
23	22	21	20	19	18	17	16
ALPHA							
15	14	13	12	11	10	9	8
ALPHA							
7	6	5	4	3	2	1	0
ALPHA							

- **ALPHA: Alpha Word x**

The four 32-bit Alpha Word registers contain the 128-bit primitive of GF(2^{128}) to use for the first processing.

58. Secure Hash Algorithm (SHA)

58.1 Description

The Secure Hash Algorithm (SHA) is compliant with the American *FIPS (Federal Information Processing Standard) Publication 180-2* specification.

The 512/1024-bit block of message is respectively stored in 16/32 x 32-bit registers, (SHA_IDATARx/SHA_IODATARx) which are write-only.

As soon as the input data is written, the hash processing may be started. The registers comprising the block of a padded message must be entered consecutively. Then the message digest is ready to be read out on the 5 up to 8/16 x 32-bit output data registers (SHA_IODATARx) or through the DMA channels.

58.2 Embedded Characteristics

- Supports Secure Hash Algorithm (SHA1, SHA224, SHA256, SHA384, SHA512)
- Supports Hash-based Message Authentication Code (HMAC) algorithm (HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512)
- Compliant with *FIPS Publication 180-2*
- Supports automatic padding of messages
- Supports up to 2 sets of initial hash values registers (HMAC acceleration or other)
- Supports automatic check of the hash (HMAC acceleration or other)
- Tightly coupled to AES for protocol layers improved performances
- Configurable Processing Period:
  - 85 Clock Cycles to obtain a fast SHA1 runtime, 88 clock cycles for SHA384, SHA512 or 209 Clock Cycles for Maximizing Bandwidth of Other Applications
  - 72 Clock Cycles to obtain a fast SHA224, SHA256 runtime or 194 Clock Cycles for Maximizing Bandwidth of Other Applications
- Connection to DMA Channel Capabilities Optimizes Data Transfers
- Double Input Buffer Optimizes Runtime

58.3 Product Dependencies

58.3.1 Power Management

The SHA may be clocked through the Power Management Controller (PMC), so the programmer must first configure the PMC to enable the SHA clock.

58.3.2 Interrupt Sources

The SHA interface has an interrupt line connected to the Interrupt Controller.

Handling the SHA interrupt requires programming the interrupt controller before configuring the SHA.

Table 58-1. Peripheral IDs

Instance	ID
SHA	12

58.4 Functional Description

The Secure Hash Algorithm (SHA) module requires a padded message according to FIPS180-2 specification.This message can be provided with the padding to the SHA module, or the padding can be automatically computed by the SHA module if the size of the message is provided. The first block of the message must be indicated to the module by a specific command. The SHA module produces an N-bit message digest each time a block is written and processing period ends, where N is 160 for SHA1, 224 for SHA224,256 for SHA256, 384 for SHA384, 512 for SHA512. The SHA module is also capable of computing Hash-based Message Authentication Code (HMAC) algorithm.

58.4.1 SHA Algorithm

The SHA can process SHA1, SHA224, SHA256, SHA384, SHA512 by configuring the ALGO field in the SHA Mode register (SHA_MR).

58.4.2 HMAC Algorithm

The HMAC algorithm is as follows:

$HMAC_K(m) = h((K_0 \oplus opad) || h((K_0 \oplus ipad) || m))$

where:

- h = SHA function
- K_0 = the key K after any necessary preprocessing to form a block size key
- m = message to authenticate
- || = concatenation operator
- $\oplus$ = XOR operator
- ipad = predefined constant (0x3636...3636)
- opad = predefined constant (0x5C5C...5C5C)

The SHA provides a fully optimized processing of the HMAC algorithm by executing the following operations:

- starting the SHA algorithm from any user predefined hash value, thus ‘$h(K_0 \oplus ipad)$’ for first HMAC hash and ‘$h(K_0 \oplus opad)$’ for second HMAC hash
- performing automatic padding
- routing automatically the first hash result ‘$h((K_0 \oplus ipad) || m)$’ to the source of the second hash processing ‘$h((K_0 \oplus opad) || (\text{first hash result}))$’ including the concatenation of the first hash result to ‘$K_0 \oplus opad$’.

To perform the HMAC operation, the ALGO field value must be greater than 7, the automatic padding feature must be enabled (MSGSIZE and BYTCNT fields differ from 0) and the SHA internal initial hash value registers 0 and 1 must be configured, respectively, with the hash results of input blocks “$K_0 \oplus ipad$” and “$K_0 \oplus opad$” (refer to Section 58.4.5 “Internal Registers for Initial Hash Value or Expected Hash Result”).

The size of the message (‘m’) must be written in the MSGSIZE and BYTCNT fields.

The FIRST bit in the SHA Control register (SHA_CR) should be set before writing the first block of the message.

The SHA can process HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512 by configuring the ALGO field in the SHA_MR.

58.4.3 Processing Period

The processing period can be configured.

The short processing period allocates bandwidth to the SHA module, whereas the long processing period allocates more bandwidth on the system bus to other applications. An example is DMA channels not associated with SHA.

In SHA1 mode, the shortest processing period is 85 clock cycles + 2 clock cycles for start command synchronization. The longest period is 209 clock cycles + 2 clock cycles.

In SHA384, SHA512 mode, the shortest processing period is 88 clock cycles + 2 clock cycles for start command synchronization. The longest period is 209 clock cycles + 2 clock cycles.

In SHA256 and SHA224 mode, the shortest processing period is 72 clock cycles + 2 clock cycles for start command synchronization. The longest period is 194 clock cycles + 2 clock cycles.

58.4.4 Double Input Buffer

The SHA Input Data registers (SHA_IDATARx) can be double-buffered to reduce the runtime of large files.

Double-buffering allows a new message block to be written while the previous message block is being processed. This is only possible when DMA accesses are performed (SMOD = 2).

The DUALBUFF bit in the SHA_MR must be set to have double input buffer access.

58.4.5 Internal Registers for Initial Hash Value or Expected Hash Result

The SHA module embeds two sets of internal registers (IR0, IR1) to store different data used by the SHA or HMAC algorithms (See Figure 58-1). These internal registers are accessed through SHA Input Data registers (SHA_IDATARx).

When the ALGO field selects SHA algorithms, IR0 can be configured with a user initial hash value. This initial hash value can be used to compute a custom hash algorithm with two sets of different initial constants, or to continue a hash computation by providing the intermediate hash value previously returned by the SHA module.

When the ALGO field selects SHA algorithms, IR1 can be configured with either a user initial hash value or an expected hash result. The expected hash result must be configured in the IR1 if the field CHECK = 1 (See Section 58.4.7 “Automatic Check”). If the field CHECK = 0 or 2, IR1 can be configured with a user initial hash value that differs from IR0 value.

When the ALGO field selects HMAC algorithms, IR0 must be configured with the hash result of $K_0 \oplus$ ipad and IR1 must be configured with the hash result of $K_0 \oplus$ opad. These precomputed first blocks speed up the HMAC computation by saving the time to compute the intermediate hash values of the first block which is constant while the secret key is constant (See Section 58.4.2 “HMAC Algorithm”).

Table 58-2. Configuration Values of Internal Registers

	SHA modes (ALGO < 8)			HMAC modes (ALGO > 7)
	CHECK = 0	CHECK = 1	CHECK = 2	
IR0	User Initial Hash	User Initial Hash	User Initial Hash	hash($K_0 \oplus$ ipad)
IR1	User Initial Hash	Expected Hash Result	User Initial Hash	hash($K_0 \oplus$ opad)

To calculate the initial HMAC values, follow this sequence:

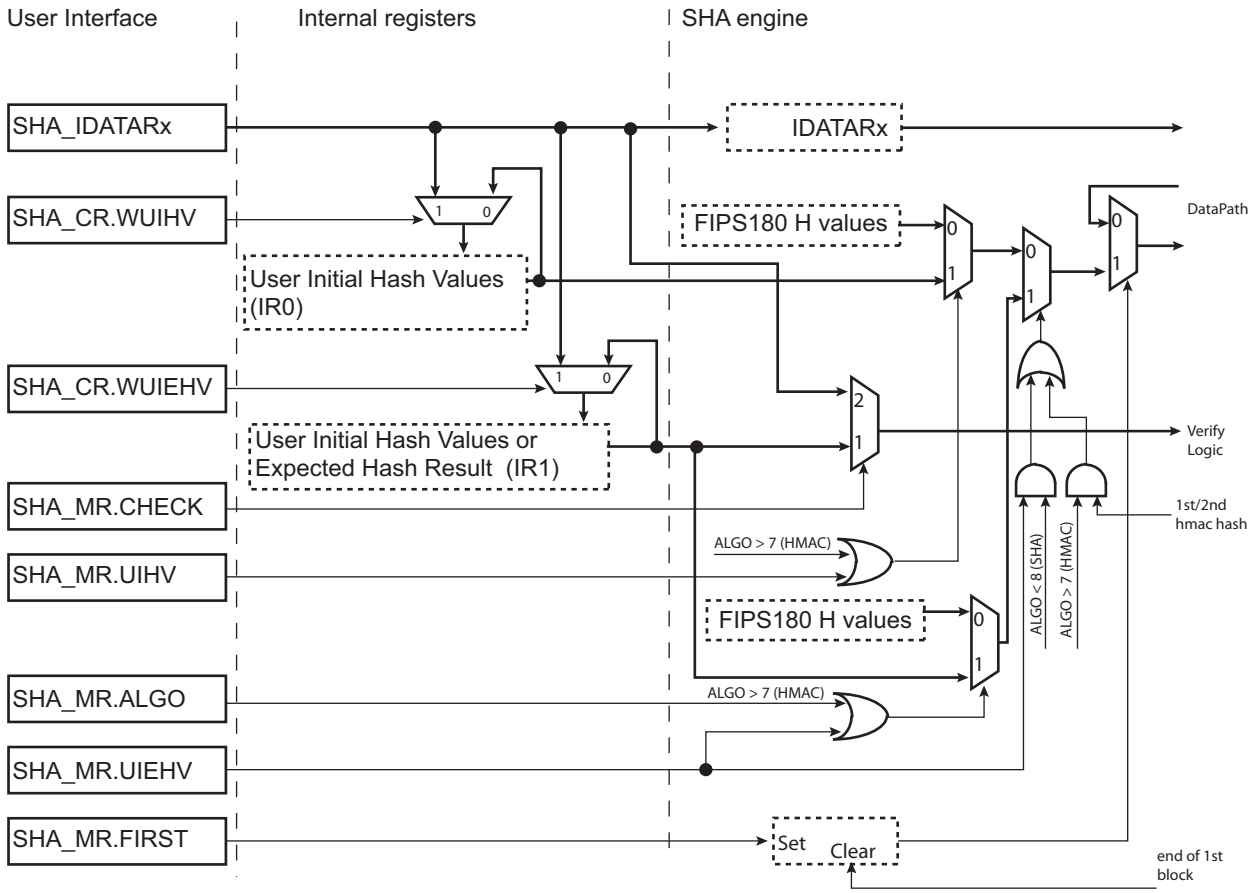
1. Calculate K_0.
2. Calculate $K_0 \oplus$ ipad and $K_0 \oplus$ opad.
3. Perform a hash of the result of $K_0 \oplus$ ipad and $K_0 \oplus$ opad (auto-padding must be disabled for that type of hash).
4. Write h($K_0 \oplus$ ipad) and h($K_0 \oplus$ opad) in IR0 and IR1 respectively.

To write IR0 or IR1, follow this sequence:

1. Set bit WUIHV (IR0) or WUIEHV (IR1) in SHA_CR
2. Write the data in SHA_IDATARx. The number of registers to write depends on the type of data (user initial hash values or expected hash result) and on the type of algorithm selected:
   - SHA_IDATAR0 to SHA_IDATAR4 for data used in algorithms based on SHA1
   - SHA_IDATAR0 to SHA_IDATAR7 for data used in algorithms based on SHA256
   - SHA_IDATAR0 to SHA_IDATAR15 for data used in algorithms based on SHA512
   - SHA_IDATAR0 to SHA_IDATAR6 for expected hash result of algorithms based on SHA224
   - SHA_IDATAR0 to SHA_IDATAR11 for expected hash result of algorithms based on SHA384
3. Clear bit WUIHV or WUIEHV in SHA_CR.

IR0 and IR1 are automatically selected for HMAC processing if the field ALGO selects HMAC algorithms. If SHA algorithms are selected, the internal registers are selected if the corresponding UIHV or UIEHV bits are set.

Figure 58-1. User Initial Hash Value and Expected Hash Internal Register Access



58.4.6 Automatic Padding

The SHA module features an automatic padding computation to speed up the execution of the algorithm.

The automatic padding function requires the following information:

- Complete message size in bytes to be written in the MSGSIZE field of the SHA Message Size register (SHA_MSR).
  The size of the message is written at the end of the last block, as required by the FIPS180-2 specification (the size is automatically converted into a bit-size).
- Number of remaining bytes (to write in the SHA_IDATARx) to be written in the BYTCNT field of the SHA Bytes Count register (SHA_BCR).
  Automatic padding occurs when the BYTCNT field reaches 0. At each write in the SHA Input registers, the BYTCNT field value is decreased by the number of bytes written.

The BYTCNT field value must be written with the same value as the MSGSIZE field value if the full message is processed. If the message is partially preprocessed and an initial hash value is used, BYTCNT must be written with the remaining bytes to hash while MSGSIZE holds the message size.

To disable the automatic padding feature, both the MSGSIZE and BYTCNT fields must be configured with 0.

58.4.7 Automatic Check

The SHA module features an automatic check of the hash result with the expected hash. A check failure can generate an interrupt if configured in the SHA Interrupt Enable register (SHA_IER).

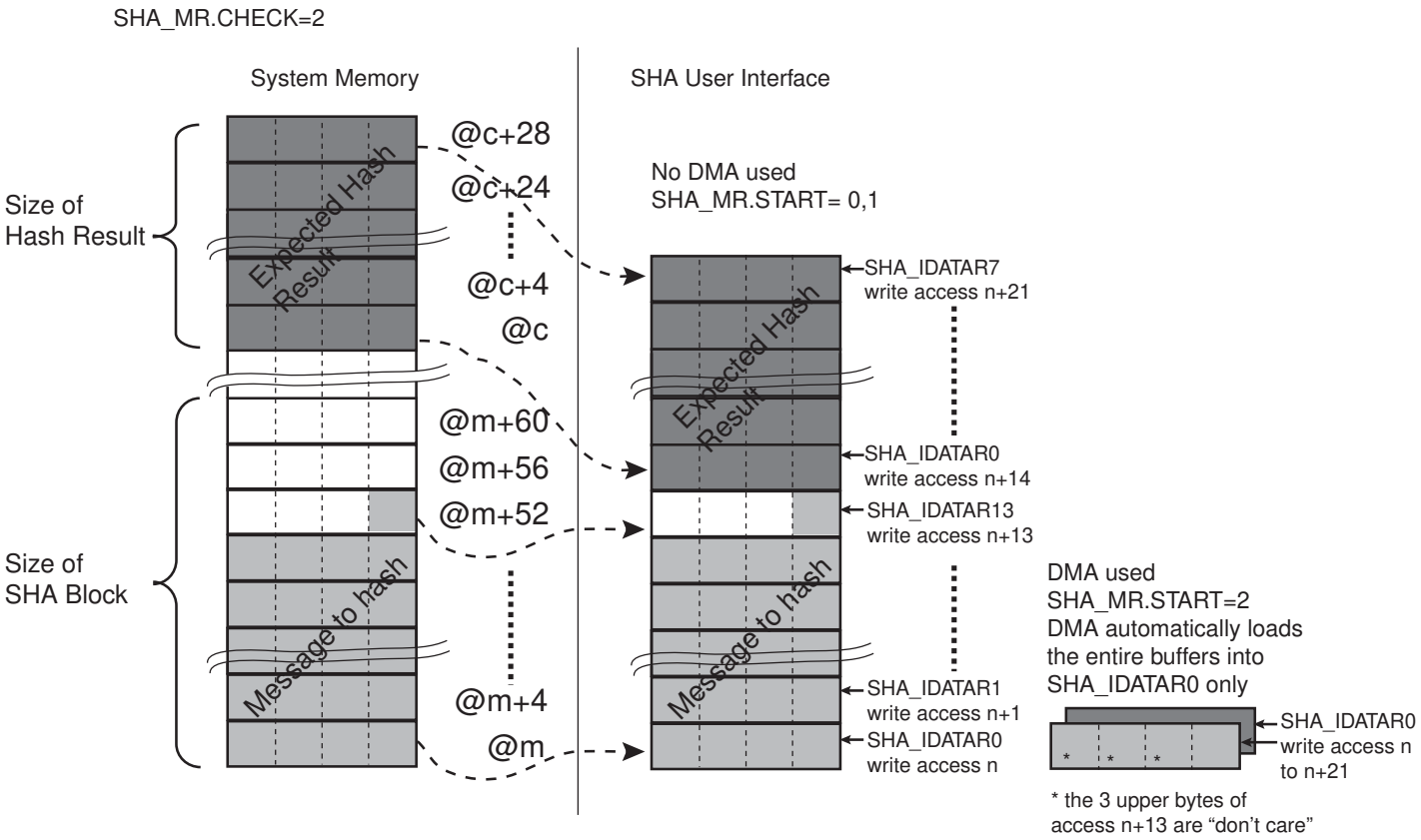
Automatic check requires the automatic padding feature to be enabled (MSGSIZE and BYTCNT fields must be greater than 0).

There are two methods to configure the expected hash result:

- if the field CHECK = 1, the expected hash result is read from the internal register (IR1). This method cannot be used when HMAC algorithms is selected because this register is already used to store user initial hash values for the second hash processing. IR1 cannot be read by software.
- If the field CHECK = 2, the expected hash result is written in the SHA_IDATARx after the message.

When CHECK = 2, the method can provide more flexibility of use if a message is stored in system memory together with its expected hash result. A DMA with linked list can be used to ease the transfer of the message and its expected hash result.

Figure 58-2. Message and Expected Hash Result Memory Mapping



The number of 32-bit words of the hash result to check with the expected hash can be selected with the CHKCNT field in the SHA_MR. The status of the check is available in the CHKST field in the SHA Interrupt Status Register (SHA_ISR).

An interrupt can be generated (if enabled) when the check is completed. The check occurs several clock cycles after the computation of the requested hash, so the interrupt and the CHECKF bit are set several clock cycles after the DATRDY flag of the SHA_ISR.

58.4.8 Protocol Layers Improved Performances

The SHA can be tightly coupled to the AES module to improve performances when processing protocol layers such as IPsec or OpenSSL.

When the AES is configured to be tightly coupled to SHA (AES_MR), SHA must be always configured in Double Buffer mode (DUALBUFF = 1 in SHA_MR).

Refer to section AES for details.

58.4.9 Start Modes

The SMOD field in the SHA_MR is used to select the Hash Processing Start mode.

58.4.9.1 Manual Mode

In Manual mode, the sequence is as follows:

1. Set the bit DATRDY (Data Ready) in the SHA_IER, depending on whether an interrupt is required at the end of processing.
2. If the initial hash values differ from the FIPS standard, set the bits UIHV and UIEHV in the SHA_MR depending on the configure the initial values.
   If the initial hash values comply with the FIPS180-2 specification, clear the bits UIHV and UIEHV in the SHA_MR.
3. If automatic padding is required, configure the MSGSIZE field in the SHA_MSR with the number of bytes of the message, and configure the BYTCNT field in the SHA_BCR with the remaining number of bytes to write. The BYTCNT field must be written with a value different from MSGSIZE field value if the message is preprocessed and completed by using user initial hash values.
   If automatic padding is not required, configure the MSGSIZE field in the SHA_MSR and the BYTCNT field in the SHA_BCR to 0.
4. For the first block of a message, the FIRST command must be set by writing a 1 into the corresponding bit of the SHA Control Register (SHA_CR). For the other blocks, there is nothing to write.
5. Write the block to be processed in the SHA_IDATARx.
6. To begin processing, set the START bit in the SHA_CR.
7. When processing is completed, the bit DATRDY in the Interrupt Status register (SHA_ISR) raises. If an interrupt has been enabled by setting the bit DATRDY in SHA_IER, the interrupt line of the SHA is activated.
8. Repeat the write procedure for each block, start procedure and wait for the interrupt procedure up to the last block of the entire message. Each time the start procedure is complete, the DATRDY flag is cleared.
9. After the last block is processed (DATRDY flag is set, if an interrupt has been enabled by setting the bit DATRDY in SHA_IER, the interrupt line of the SHA is activated), read the message digest in the Output Data Registers. The DATRDY flag is automatically cleared when reading the SHA_IODATARx registers.

58.4.9.2 Auto Mode

In Auto mode, processing starts as soon as the correct number of SHA_IDATARx is written. No action in the SHA_CR is necessary.

58.4.9.3 DMA Mode

The DMA can be used in association with the SHA to perform the algorithm on a complete message without any action by the software during processing.

The SMOD field in SHA_MR must be configured to 2.

The DMA must be configured with non-incremental addresses.

The start address of any transfer descriptor must be set to point to the SHA_IDATAR0.

The DMA chunk size must be set to transfer, for each trigger request, 16 words of 32 bits.

The FIRST bit of the SHA_CR must be set before starting the DMA when the first block is transferred.

The DMA generates an interrupt when the end of buffer transfer is completed but the SHA processing is still in progress. The end of SHA processing is indicated by the flag DATRDY in the SHA_SR.
If automatic padding is disabled, the end of SHA processing requires two interrupts to be verified. The DMA end of transfer interrupt must be verified first, then the SHA DATRDY interrupt must be enabled and verified (see Figure 58-3).

If automatic padding is enabled, the end of SHA processing requires only one interrupt to be verified (see Figure 58-4). The DMA end of transfer is not required, so the SHA DATRDY interrupt must be enabled prior to start the DMA and DATRDY interrupt is the only one to be verified.

Figure 58-3. interrupts Processing with DMA

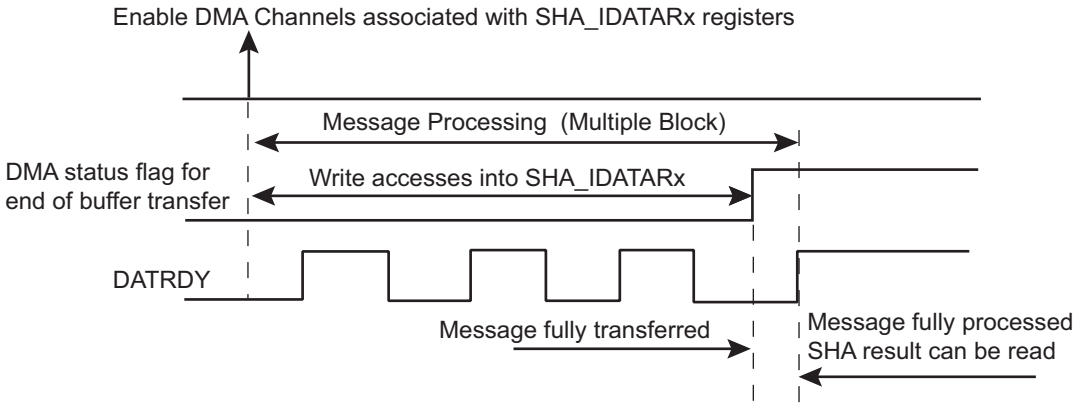
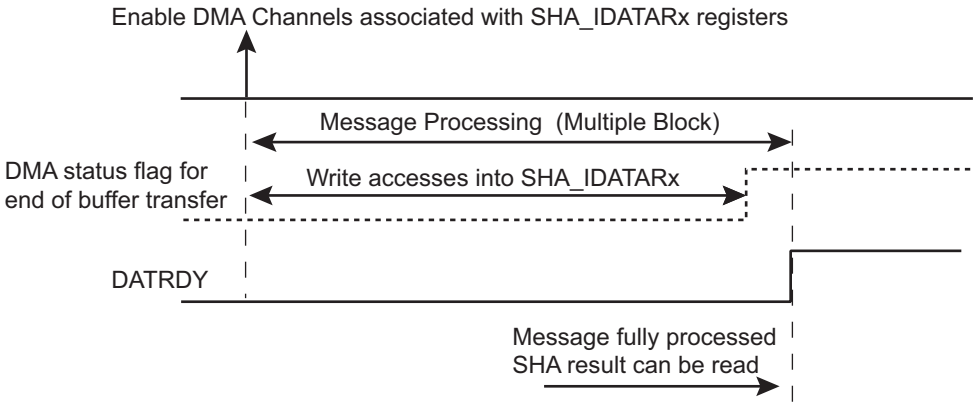


Figure 58-4. interrupts processing with DMA and automatic padding



58.4.9.4 SHA Register Endianism

In ARM processor-based products, the system bus and processors manipulate data in little-endian form. The SHA interface requires little-endian format words. However, in accordance with the protocol of FIPS 180-2 specification, data is collected, processed and stored by the SHA algorithm in big-endian form.

The following example illustrates how to configure the SHA:

If the first 64 bits of a message (according to FIPS 180-2, i.e., big-endian format) to be processed is 0xcafedeca_01234567, then the SHA_IDATAR0 and SHA_IDATAR1 registers must be written with the following pattern:

- SHA_IDATAR0 = 0xcadefeca
- SHA_IDATAR1 = 0x67452301

In a little-endian system, the message (according to FIPS 180-2) starting with pattern 0xcafedeca_01234567 is stored into memory as follows:

- 0xca stored at initial offset (for example 0x00),
- then 0xfe stored at initial offset + 1 (i.e., 0x01),
- 0xde stored at initial offset + 2 (i.e., 0x02),
- 0xca stored at initial offset + 3 (i.e., 0x03).

If the message is received through a serial-to-parallel communication channel, the first received character is 0xca and it is stored at the first memory location (initial offset). The second byte, 0xfe, is stored at initial offset + 1.

When reading on a 32-bit little-endian system bus, the first word read back from system memory is 0xcadefeca.

When the SHA_IODATARx registers are read, the hash result is organized in little-endian format, allowing system memory storage in the same format as the message.

Taking an example from the FIPS 180-2 specification Appendix B.1, the endianism conversion can be observed.

For this example, the 512-bit message is:

0x61626380000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000018

and the expected SHA-256 result is:

0xba7816bf_8f01cfea_414140de_5dae2223_b00361a3_96177a9c_b410ff61_f20015ad

If the message has not already been stored in the system memory, the first step is to convert the input message to little-endian before writing to the SHA_IDATARx registers. This would result in a write of:

SHA_IDATAR0 = 0x80636261...... SHA_IDATAR15 = 0x18000000

The data in the output message digest registers, SHA_IODATARx, contain SHA_IODATAR0 = 0xbf1678ba... SHA_IODATAR7 = 0xad1500f2 which is the little-endian format of 0xba7816bf,..., 0xf20015ad.

Reading SHA_IODATAR0 to SHA_IODATAR1 and storing into a little-endian memory system forces hash results to be stored in the same format as the message.

When the output message is read, the user can convert back to big-endian for a resulting message value of:

0xba7816bf_8f01cfea_414140de_5dae2223_b00361a3_96177a9c_b410ff61_f20015ad

58.4.10 Security Features

When an unspecified register access occurs, the URAD bit in the SHA_ISR is set. Its source is then reported in the Unspecified Register Access Type field (URAT). Only the last unspecified register access is available through the URAT field.

Several kinds of unspecified register accesses can occur:

- SHA_IDATARx written during data processing in DMA mode
- SHA_IODATARx read during data processing
- SHA_MR written during data processing
- Write-only register read access

The URAD bit and the URAT field can only be reset by the SWRST bit in the SHA_CR.

58.5 Secure Hash Algorithm (SHA) User Interface

Table 58-3. Register Mapping

Offset	Register	Name	Access	Reset
0x00	Control Register	SHA_CR	Write-only	–
0x04	Mode Register	SHA_MR	Read/Write	0x0000100
0x08–0x0C	Reserved	–	–	–
0x10	Interrupt Enable Register	SHA_IER	Write-only	–
0x14	Interrupt Disable Register	SHA_IDR	Write-only	–
0x18	Interrupt Mask Register	SHA_IMR	Read-only	0x0
0x1C	Interrupt Status Register	SHA_ISR	Read-only	0x0
0x20	Message Size Register	SHA_MSR	Read/Write	0x0
0x24–0x2C	Reserved	–	–	–
0x30	Bytes Count Register	SHA_BCR	Read/Write	0x0
0x34–0x3C	Reserved	–	–	–
0x40	Input Data 0 Register	SHA_IDATAR0	Write-only	–
...	...	...	...	...
0x7C	Input Data 15 Register	SHA_IDATAR15	Write-only	–
0x80	Input/Output Data 0 Register	SHA_IODATAR0	Read/Write	0x0
...	...	...	...	...
0x9C	Input/Output Data 7 Register	SHA_IODATAR7	Read/Write	0x0
0xA0	Input/Output Data 8 Register	SHA_IODATAR8	Read/Write	0x0
...	...	...	...	...
0xBC	Input/Output Data 15 Register	SHA_IODATAR15	Read/Write	0x0
0xC0–0xFC	Reserved	–	–	–

58.5.1 SHA Control Register

Name: SHA_CR

Address: 0xF0028000

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	WUIEHV	WUIHV	–	–	–	SWRST
7	6	5	4	3	2	1	0
–	–	–	FIRST	–	–	–	START

- **START: Start Processing**

0: No effect.

1: Starts manual hash algorithm process.

- **FIRST: First Block of a Message**

0: No effect.

1: Indicates that the next block to process is the first one of a message.

- **SWRST: Software Reset**

0: No effect.

1: Resets the SHA. A software-triggered hardware reset of the SHA interface is performed.

- **WUIHV: Write User Initial Hash Values**

0: SHA_IDATARx accesses are routed to the data registers.

1: SHA_IDATARx accesses are routed to the internal registers (IR0).

- **WUIEHV: Write User Initial or Expected Hash Values**

0: SHA_IDATARx accesses are routed to the data registers.

1: SHA_IDATARx accesses are routed to the internal registers (IR1).

58.5.2 SHA Mode Register

Name: SHA_MR

Address: 0xF0028004

Access: Read/Write

31	30	29	28	27	26	25	24
CHKCNT				–	–	CHECK	
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	DUALBUFF
15	14	13	12	11	10	9	8
–	–	–	–	ALGO			
7	6	5	4	3	2	1	0
–	UIEHV	UIHV	PROCDLY	–	–	SMOD	

- **SMOD: Start Mode**

Value	Name	Description
0	MANUAL_START	Manual Mode
1	AUTO_START	Auto Mode
2	IDATAR0_START	SHA_IDATAR0 access only Auto Mode

Values not listed in the table must be considered as “reserved”.

If a DMA transfer is used, configure the SMOD value with 1 or 2. Refer to Section 58.4.9.3 “DMA Mode” for more details.

- **PROCDLY: Processing Delay**

Value	Name	Description
0	SHORTEST	SHA processing runtime is the shortest one
1	LONGEST	SHA processing runtime is the longest one (reduces the SHA bandwidth requirement, reduces the system bus overload)

When SHA1 algorithm is processed, runtime period is either 85 or 209 clock cycles.

When SHA256 or SHA224 algorithm is processed, runtime period is either 72 or 194 clock cycles.

When SHA384 or SHA512 algorithm is processed, runtime period is either 88 or 209 clock cycles.

- **UIHV: User Initial Hash Value Registers**

0: The SHA algorithm is started with the standard initial values as defined in the FIPS180-2 specification.

1: The SHA algorithm is started with the user initial hash values stored in the internal register 0 (IR0). If HMAC is configured, UIHV has no effect (i.e. IR0 is selected).

- **UIEHV: User Initial or Expected Hash Value Registers**

0: The SHA algorithm is started with the standard initial values as defined in the FIPS180-2 specification.

1: The SHA algorithm is started with the user initial hash values stored in the internal register 1 (IR1). If HMAC is configured, UIEHV has no effect (i.e. IR1 is always selected).

- **ALGO: SHA Algorithm**

Value	Name	Description
0	SHA1	SHA1 algorithm processed
1	SHA256	SHA256 algorithm processed
2	SHA384	SHA384 algorithm processed
3	SHA512	SHA512 algorithm processed
4	SHA224	SHA224 algorithm processed
8	HMAC_SHA1	HMAC algorithm with SHA1 Hash processed
9	HMAC_SHA256	HMAC algorithm with SHA256 Hash processed
10	HMAC_SHA384	HMAC algorithm with SHA384 Hash processed
11	HMAC_SHA512	HMAC algorithm with SHA512 Hash processed
12	HMAC_SHA224	HMAC algorithm with SHA224 Hash processed

Values not listed in the table must be considered as “reserved”.

- **DUALBUFF: Dual Input Buffer**

Value	Name	Description
0	INACTIVE	SHA_IDATARx and SHA_IODATARx cannot be written during processing of previous block.
1	ACTIVE	SHA_IDATARx and SHA_IODATARx can be written during processing of previous block when SMOD value = 2. It speeds up the overall runtime of large files.

- **CHECK: Hash Check**

Value	Name	Description
0	NO_CHECK	No check is performed
1	CHECK_EHV	Check is performed with expected hash stored in internal expected hash value registers.
2	CHECK_MESSAGE	Check is performed with expected hash provided after the message.

Values not listed in table must be considered as “reserved”.

- **CHKCNT: Check Counter**

Number of 32-bit words to check. The value 0 indicates that the number of words to compare will be based on the algorithm selected (5 words for SHA1, 7 words for SHA224, 8 words for SHA256, 12 words for SHA384, 16 words for SHA512).

58.5.3 SHA Interrupt Enable Register

Name: SHA_IER

Address: 0xF0028010

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	CHECKF
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	URAD
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

- **DATRDY: Data Ready Interrupt Enable**
- **URAD: Unspecified Register Access Detection Interrupt Enable**
- **CHECKF: Check Done Interrupt Enable**

58.5.4 SHA Interrupt Disable Register

Name: SHA_IDR

Address: 0xF0028014

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	CHECKF
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	URAD
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

- **DATRDY: Data Ready Interrupt Disable**
- **URAD: Unspecified Register Access Detection Interrupt Disable**
- **CHECKF: Check Done Interrupt Disable**

58.5.5 SHA Interrupt Mask Register

Name: SHA_IMR

Address: 0xF0028018

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	CHECKF
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	URAD
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

- **DATRDY: Data Ready Interrupt Mask**
- **URAD: Unspecified Register Access Detection Interrupt Mask**
- **CHECKF: Check Done Interrupt Mask**

58.5.6 SHA Interrupt Status Register

Name: SHA_ISR

Address: 0xF002801C

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
CHKST				–	–	–	CHECKF
15	14	13	12	11	10	9	8
	URAT			–	–	–	URAD
7	6	5	4	3	2	1	0
–	–	–	WRDY	–	–	–	DATRDY

- **DATRDY: Data Ready (cleared by writing a 1 to bit SWRST or START in SHA_CR, or by reading SHA_IODATARx)**

0: Output data is not valid.

1: 512/1024-bit block process is completed.

DATRDY is cleared when one of the following conditions is met:

- Bit START in SHA_CR is set.
- Bit SWRST in SHA_CR is set.
- The hash result is read.

- **WRDY: Input Data Register Write Ready**

0: SHA_IDATAR0 cannot be written

1: SHA_IDATAR0 can be written

- **URAD: Unspecified Register Access Detection Status (cleared by writing a 1 to SWRST bit in SHA_CR)**

0: No unspecified register access has been detected since the last SWRST.

1: At least one unspecified register access has been detected since the last SWRST.

- **URAT: Unspecified Register Access Type (cleared by writing a 1 to SWRST bit in SHA_CR)**

Value	Description
0	SHA_IDATAR0 to SHA_IDATAR15 written during the data processing in DMA mode (URAD = 1 and URAT = 0 can occur only if DUALBUFF is cleared in SHA_MR).
1	Output Data Register read during the data processing.
2	SHA_MR written during the data processing.
3	Write-only register read access.

Only the last Unspecified Register Access Type is available through the URAT field.

- **CHECKF: Check Done Status (cleared by writing START or SWRST bits in SHA_CR or by reading SHA_IODATARx)**

0: Hash check has not been computed.

1: Hash check has been computed, status is available in the CHKST bits.

- **CHKST: Check Status (cleared by writing START or SWRST bits in SHA_CR or by reading SHA_IODATARx)**

The value 5 indicates identical hash values (expected hash = hash result). Any other value indicates different hash values.

58.5.7 SHA Message Size Register

Name: SHA_MSR

Address: 0xF0028020

Access: Read/Write

31	30	29	28	27	26	25	24
MSGSIZE							
23	22	21	20	19	18	17	16
MSGSIZE							
15	14	13	12	11	10	9	8
MSGSIZE							
7	6	5	4	3	2	1	0
MSGSIZE							

- **MSGSIZE: Message Size**

The size in bytes of the message. When MSGSIZE differs from 0, the SHA appends the corresponding value converted in bits after the padding section, as described in the FIPS180-2 specification.

To disable automatic padding, MSGSIZE field must be written to 0.

58.5.8 SHA Bytes Count Register

Name: SHA_BCR

Address: 0xF0028030

Access: Read/Write

31	30	29	28	27	26	25	24
BYTCNT							

23	22	21	20	19	18	17	16
BYTCNT							

15	14	13	12	11	10	9	8
BYTCNT							

7	6	5	4	3	2	1	0
BYTCNT							

- **BYTCNT: Remaining Byte Count Before Auto Padding**

When the hash processing starts from the beginning of a message (without preprocessed hash part), BYTCNT must be written with the same value as the MSGSIZE. If a part of the message has been already hashed and the hash does not start from the beginning, BYTCNT must be configured with the number of bytes remaining to process before padding section.

When read, provides the size in bytes of message remaining to be written before the automatic padding starts.

BYTCNT field is automatically updated each time a write occurs in the SHA_IDATARx and SHA_IODATARx.

When BYTCNT reaches 0, the MSGSIZE is converted into bit count and appended at the end of the message after the padding as described in the FIPS180-2 specification.

To disable automatic padding, MSGSIZE and BYTCNT fields must be written to 0.

58.5.9 SHA Input Data x Register

Name: SHA_IDATARx [x=0..15]

Address: 0xF0028040

Access: Write-only

31	30	29	28	27	26	25	24
IDATA							
23	22	21	20	19	18	17	16
IDATA							
15	14	13	12	11	10	9	8
IDATA							
7	6	5	4	3	2	1	0
IDATA							

- **IDATA: Input Data**

The 32-bit Input Data registers allow to load the data block used for hash processing.

These registers are write-only to prevent the input data from being read by another application.

SHA_IDATAR0 corresponds to the first word of the block, SHA_IDATAR15 to the last word of the last block in case SHA algorithm is set to SHA1, SHA224, SHA256 or SHA_IODATA15R to the last word of the block if SHA algorithm is SHA384 or SHA512 (refer to Section 58.5.10 “SHA Input/Output Data Register x”).

58.5.10 SHA Input/Output Data Register x

Name: SHA_IODATARx [x=0..15]

Address: 0xF0028080

Access: Read/Write

31	30	29	28	27	26	25	24
IODATA							
23	22	21	20	19	18	17	16
IODATA							
15	14	13	12	11	10	9	8
IODATA							
7	6	5	4	3	2	1	0
IODATA							

- **IODATA: Input/Output Data**

These registers can be used to read the resulting message digest and to write the second part of the message block when the SHA algorithm is SHA-384 or SHA-512.

SHA_IODATA0R to SHA_IODATA15R can be written or read but reading these offsets does not return the content of corresponding parts (words) of the message block. Only results from SHA calculation can be read through these registers.

When SHA processing is in progress, these registers return 0x0000.

SHA_IODATAR0 corresponds to the first word of the message digest; SHA_IODATAR4 to the last one in SHA1 mode, SHA_ODATAR6 in SHA224, SHA_IODATAR7 in SHA256, SHA_IODATAR11 in SHA384 or SHA_IODATAR15 in SHA512.

When SHA224 is selected, the content of SHA_ODATAR7 must be ignored.

When SHA384 is selected, the content of SHA_IODATAR12 to SHA_IODATAR15 must be ignored.

59. Triple Data Encryption Standard (TDES)

59.1 Description

The Triple Data Encryption Standard (TDES) is compliant with the American *FIPS (Federal Information Processing Standard) Publication 46-3* specification.

The TDES supports the four different confidentiality modes of operation (ECB, CBC, OFB and CFB), specified in the *FIPS (Federal Information Processing Standard) Publication 81* and is compatible with the Peripheral Data Controller channels for all of these modes, minimizing processor intervention for large buffer transfers.

The 64-bit long keys and input data (and initialization vector for some modes) are each stored in two corresponding 32-bit write-only registers:

Key x Word Registers TDES_KEYxWR0 and TDES_KEYxWR1

Input Data Registers TDES_IDATAR0 and TDES_IDATAR1

Initialization Vector Registers TDES_IVR0 and TDES_IVR1

As soon as the initialization vector, the input data and the key are configured, the encryption/decryption process may be started. Then the encrypted/decrypted data is ready to be read out on the two 32-bit Output Data registers (TDES_ODATARx) or through the DMA channels.

59.2 Embedded Characteristics

- Supports Single Data Encryption Standard (DES) and Triple Data Encryption Algorithm (TDEA or TDES)
- Compliant with *FIPS Publication 46-3, Data Encryption Standard (DES)*
- 64-bit Cryptographic Key for TDES
- Two-key or Three-key Algorithms for TDES
- 18-clock Cycles Encryption/Decryption Processing Time for DES
- 50-clock Cycles Encryption/Decryption Processing Time for TDES
- Supports eXtended Tiny Encryption Algorithm (XTEA)
- 128-bit key for XTEA and Programmable Round Number up to 64
- Supports the Four Standard Modes of Operation specified in the *FIPS Publication 81, DES Modes of Operation*
  - Electronic Code Book (ECB)
  - Cipher Block Chaining (CBC)
  - Cipher Feedback (CFB)
  - Output Feedback (OFB)
- 8-, 16-, 32- and 64-bit Data Sizes Possible in CFB Mode
- Last Output Data Mode Allowing Optimized Message (Data) Authentication Code (MAC) Generation
- Connection to DMA Optimizes Data Transfers for all Operating Modes

59.3 Product Dependencies

59.3.1 Power Management

The TDES may be clocked through the Power Management Controller (PMC), so the programmer must first configure the PMC to enable the TDES clock.

59.3.2 Interrupt Sources

The TDES interface has an interrupt line connected to the interrupt controller. Handling the TDES interrupt requires programming the interrupt controller before configuring the TDES.

Table 59-1. Peripheral IDs

Instance	ID
TDES	11

59.4 Functional Description

The Data Encryption Standard (DES) and the Triple Data Encryption Algorithm (TDES) specify FIPS-approved cryptographic algorithms that can be used to protect electronic data. The TDES bit in the TDES Mode Register (TDES_MR) is used to select either the single DES or the Triple DES mode.

Encryption (enciphering) converts data to an unintelligible form called ciphertext. Decrypting (deciphering) the ciphertext converts the data back into its original form, called plaintext. The CIPHER bit in TDES_MR is used to choose between encryption and decryption.

A DES is capable of using cryptographic keys of 64 bits to encrypt and decrypt data in blocks of 64 bits. This 64-bit key is defined in the Key 1 Word Registers (TDES_KEY1WRx).

A TDES key consists of three DES keys, which is also referred to as a key bundle. These three 64-bit keys are defined, respectively, in the Key 1, 2 and 3 Word Registers (TDES_KEY1WRx, TDES_KEY2WRx and TDES_KEY3WRx). In Triple DES mode (TDESMOD = 1 in TDES_MR), the KEYMOD bit in TDES_MR is used to choose between a two- and a three-key algorithm, as summarized in Table 59-2.

Table 59-2. TDES Algorithms Summary

Algorithm	Mode	Data Processing Sequence Steps		
		First	Second	Third
Three-key	Encryption	Encryption with Key 1	Decryption with Key 2	Encryption with Key 3
	Decryption	Decryption with Key 3	Encryption with Key 2	Decryption with Key 1
Two-key	Encryption	Encryption with Key 1	Decryption with Key 2	Encryption with Key 1
	Decryption	Decryption with Key 1	Encryption with Key 2	Decryption with Key 1

The input to the encryption processes of the CBC, CFB, and OFB modes includes, in addition to the plaintext, a 64-bit data block called the initialization vector (IV), which must be set in the Initialization Vector Registers (TDES_IVRx). The initialization vector is used in an initial step in the encryption of a message and in the corresponding decryption of the message.

The XTEA algorithm can be used instead of DES/TDES by configuring the TDESMOD field in TDES_MR with the appropriate value 0x2. An XTEA key consists of a 128-bit key. They are defined in the Key 1 and 2 Word Registers (TDES_KEY1WRx, TDES_KEY2WRx).

The number of rounds of XTEA is defined in TDES_XTEA_RNDR and can be programmed up to 64 (1 round = 2 Feistel network rounds).

All the start and operating modes of the TDES algorithm can be applied to the XTEA algorithm.

59.4.1 Operating Modes

The TDES supports the following operating modes:

- ECB—Electronic Code Book
- CBC—Cipher Block Chaining

- OFB—Output Feedback
- CFB—Cipher Feedback
  - CFB8 (CFB where the length of the data segment is 8 bits)
  - CFB16 (CFB where the length of the data segment is 16 bits)
  - CFB32 (CFB where the length of the data segment is 32 bits)
  - CFB64 (CFB where the length of the data segment is 64 bits)

The data preprocessing, post-processing and data chaining for each mode are automatically performed. Refer to the *FIPS Publication 81* for more complete information.

These modes are selected by setting the OPMOD field in TDES_MR.

In CFB mode, four data sizes are possible (8, 16, 32 and 64 bits), configurable by means of the CFBS field in TDES_MR (see Section 59.5.2 “TDES Mode Register”).

59.4.2 Start Modes

The SMOD field in TDES_MR selects the Encryption (or Decryption) start mode.

59.4.2.1 Manual Mode

The sequence is as follows:

1. Write the TDES_MR register with all required fields, including but not limited to SMOD and OPMOD.
2. Write the 64-bit key(s) in the different Key Word Registers (TDES_KEYxWRx), depending on whether one, two or three keys are required.
3. Write the initialization vector (or counter) in the Initialization Vector Registers (TDES_IVRx).

Note: The Initialization Vector Registers concern all modes except ECB.

4. Set the bit DATRDY (Data Ready) in the TDES Interrupt Enable register (TDES_IER), depending on whether an interrupt is required or not at the end of processing.
5. Write the data to be encrypted/decrypted in the authorized Input Data Registers (see Table 59-3).

Note: In 32-, 16- and 8-bit CFB modes, writing to TDES_IDATAR1 is not allowed and may lead to processing errors.

6. Set the START bit in the TDES Control Register (TDES_CR) to begin the encryption or decryption process.
7. When the processing completes, the bit DATRDY in the TDES Interrupt Status Register (TDES_ISR) rises. If an interrupt has been enabled by setting the bit DATRDY in TDES_IER, the interrupt line of the TDES is activated.
8. When the software reads one of the Output Data Registers (TDES_ODATARx), the DATRDY bit is automatically cleared.

Table 59-3. Authorized Input Data Registers

Operating Mode	Input Data Registers to Write
ECB	All
CBC	All
OFB	All
CFB 64-bit	All
CFB 32-bit	TDES_IDATAR0
CFB 16-bit	TDES_IDATAR0
CFB 8-bit	TDES_IDATAR0

59.4.2.2 Auto Mode

The Auto Mode is similar to the Manual Mode, except that as soon as the correct number of Input Data registers is written, processing is automatically started without any action in TDES_CR.

59.4.2.3 DMA Mode

The DMA Controller can be used in association with the TDES to perform an encryption/decryption of a buffer without any action by the software during processing.

The SMOD field of TDES_MR must be set to 0x2 and the DMA must be configured with non-incremental addresses.

The start address of any transfer descriptor must be set in TDES_IDATAR0.

The DMA chunk size configuration depends on the TDES mode of operation and is listed in Table 59-4.

When writing data to TDES with the first DMA channel, data will be fetched from a memory buffer (source data). It is recommended to configure the size of source data to “words” even for CFB modes. On the contrary, the destination data size depends on the mode of operation. When reading data from the TDES with the second DMA channel, the source data is the data read from TDES and data destination is the memory buffer. In this case, source data size depends on the TDES mode of operation and is listed in Table 59-4.

Table 59-4. DMA Data Transfer Type for the Different Operating Modes

Operating Mode	Chunk Size	Destination/Source Data Transfer Type
ECB	1	Word
CBC	1	Word
OFB	1	Word
CFB 64-bit	1	Word
CFB 32-bit	1	Word
CFB 16-bit	1	Half-word
CFB 8-bit	1	Byte

59.4.3 Last Output Data Mode

This mode is used to generate cryptographic checksums on data (MAC) using a CBC-MAC or a CFB encryption algorithm (See *FIPS Publication 81 Appendix F*).

After each end of encryption/decryption, the output data is available either on the output data registers for Manual and Auto modes or at the address specified in the receive buffer pointer for DMA mode (See Table 59-5 “Last Output Data Mode Behavior versus Start Modes”).

The Last Output Data bit (LOD) in TDES_MR can be used to retrieve only the last data of several encryption/decryption processes.

This data is only available on the Output Data Registers (TDES_ODATARx).

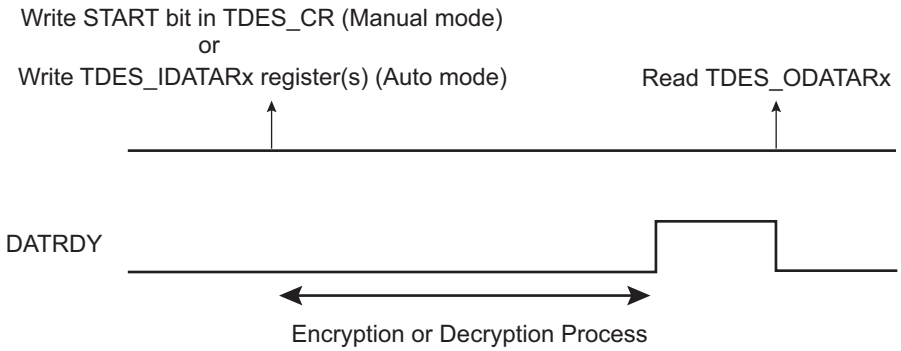
Therefore, there is no need to define a read buffer in DMA mode.

59.4.3.1 Manual and Auto Modes

TDES_MR.LOD = 0

The DATRDY flag is cleared when at least one of the Output Data Registers is read. See Figure 59-1.

Figure 59-1. Manual and Auto Modes with LOD = 0

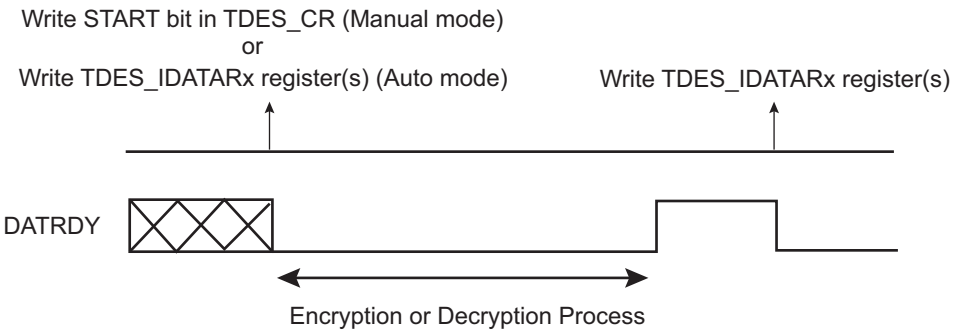


If the user does not want to read the output data registers between each encryption/decryption, the DATRDY flag will not be cleared. If the DATRDY flag is not cleared, the user will not be informed of the end of the encryptions/decryptions that follow.

TDES_MR.LOD = 1

The DATRDY flag is cleared when at least one Input Data Register is written, before the start of a new transfer. See Figure 59-2. No further Output Data Register reads are necessary between consecutive encryptions/decryptions.

Figure 59-2. Manual and Auto Modes with LOD = 1



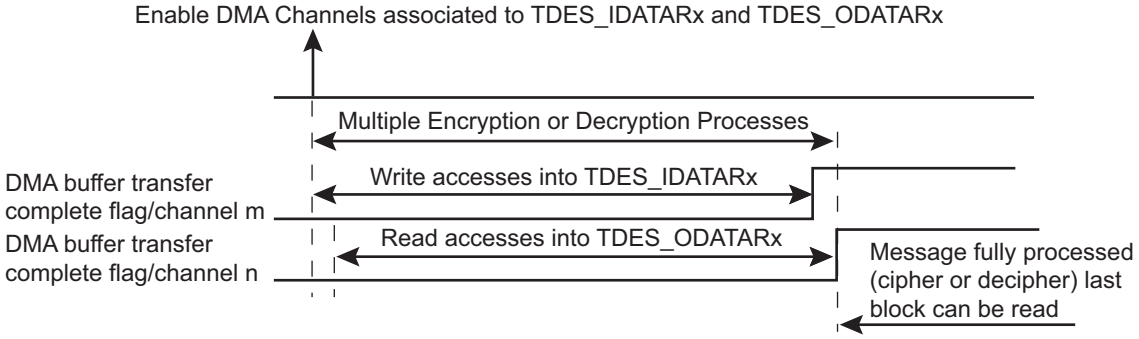
59.4.3.2 DMA Mode

TDES_MR.LOD = 0

This mode may be used for all TDES operating modes except CBC-MAC where LOD = 1 mode is recommended.

The end of the encryption/decryption is indicated by the end of DMA transfer associated to TDES_ODATARx (see Figure 59-3). Two DMA channels are required: one for writing message blocks to TDES_IDATARx and one to obtain the result from TDES_ODATARx.

Figure 59-3. DMA Transfer with LOD = 0



TDES_MR.LOD = 1

This mode is optimized to process the TDES CBC-MAC operating mode.

The user must first wait for the DMA buffer transfer complete flag, then for the flag DATRDY to rise to ensure that the encryption/decryption is completed (see Figure 59-4).

In this case, no receive buffers are required.

The output data is only available on TDES_ODATARx.

Figure 59-4. DMA Transfer with LOD = 1

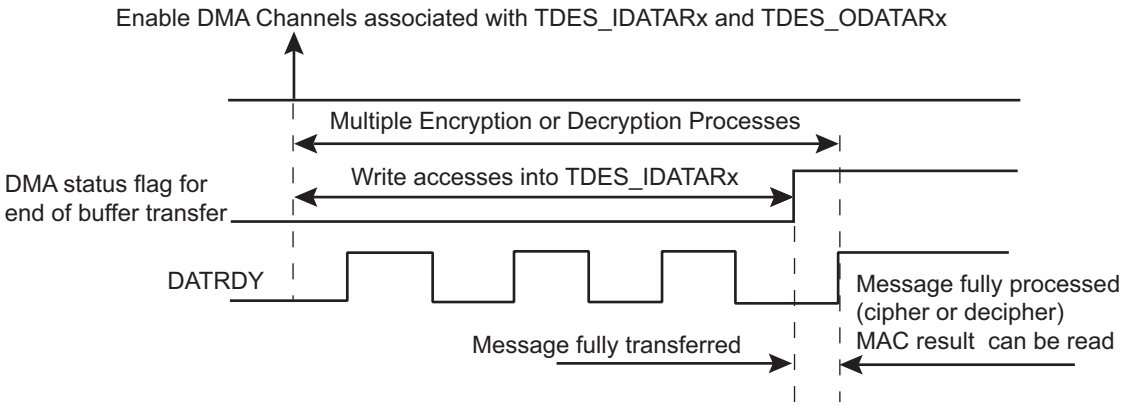


Table 59-5 summarizes the different cases.

Table 59-5. Last Output Data Mode Behavior versus Start Modes

Sequence	Manual and Auto Modes		DMA Transfer	
	LOD = 0	LOD = 1	LOD = 0	LOD = 1
DATRDY Flag Clearing Condition (1)	At least one Output Data Register must be read	At least one Input Data Register must be written	Not used	Managed by the DMA
End of Encryption/Decryption	DATRDY	DATRDY	2 DMA Buffer transfer complete flags (channel m and channel n)	DMA buffer transfer complete flag, then TDES DATRDY flag
Encrypted/Decrypted Data Result Location	In the Output Data Registers	In the Output Data Registers	Not available	In the Output Data Registers

Note: 1. Depending on the mode, there are other ways of clearing the DATRDY flag. See: Section 59.5.6 “TDES Interrupt Status Register”.

Warning: In DMA mode, reading to the Output Data registers before the last data transfer may lead to unpredictable results.

59.4.4 Security Features

59.4.4.1 Unspecified Register Access Detection

When an unspecified register access occurs, the URAD bit in TDES_ISR is set. Its source is then reported in the Unspecified Register Access Type field (URAT). Only the last unspecified register access is available through the URAT field.

Several kinds of unspecified register accesses can occur:

- Input Data Register written during the data processing in DMA mode
- Output Data Register read during the data processing
- Mode Register written during the data processing
- Write-only register read access

The URAD bit and the URAT field can only be reset by the SWRST bit in TDES_CR.

59.5 Triple Data Encryption Standard (TDES) User Interface

Table 59-6. Register Mapping

Offset	Register	Name	Access	Reset
0x00	Control Register	TDES_CR	Write-only	–
0x04	Mode Register	TDES_MR	Read/Write	0x2
0x08–0x0C	Reserved	–	–	–
0x10	Interrupt Enable Register	TDES_IER	Write-only	–
0x14	Interrupt Disable Register	TDES_IDR	Write-only	–
0x18	Interrupt Mask Register	TDES_IMR	Read-only	0x0
0x1C	Interrupt Status Register	TDES_ISR	Read-only	0x0000001E
0x20	Key 1 Word Register 0	TDES_KEY1WR0	Write-only	–
0x24	Key 1 Word Register 1	TDES_KEY1WR1	Write-only	–
0x28	Key 2 Word Register 0	TDES_KEY2WR0	Write-only	–
0x2C	Key 2 Word Register 1	TDES_KEY2WR1	Write-only	–
0x30	Key 3 Word Register 0	TDES_KEY3WR0	Write-only	–
0x34	Key 3 Word Register 1	TDES_KEY3WR1	Write-only	–
0x38–0x3C	Reserved	–	–	–
0x40	Input Data Register 0	TDES_IDATAR0	Write-only	–
0x44	Input Data Register 1	TDES_IDATAR1	Write-only	–
0x48–0x4C	Reserved	–	–	–
0x50	Output Data Register 0	TDES_ODATAR0	Read-only	0x0
0x54	Output Data Register 1	TDES_ODATAR1	Read-only	0x0
0x58–0x5C	Reserved	–	–	–
0x60	Initialization Vector Register 0	TDES_IVR0	Write-only	–
0x64	Initialization Vector Register 1	TDES_IVR1	Write-only	–
0x68–0x6C	Reserved	–	–	–
0x70	XTEA Rounds Register	TDES_XTEA_RNDR	Read/Write	0x0
0x74–0xFC	Reserved	–	–	–

59.5.1 TDES Control Register

Name: TDES_CR

Address: 0xFC044000

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	SWRST
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	START

- **START: Start Processing**

0: No effect

1: Starts Manual encryption/decryption process.

- **SWRST: Software Reset**

0: No effect

1: Resets the TDES. A software triggered hardware reset of the TDES interface is performed.

59.5.2 TDES Mode Register

Name: TDES_MR

Address: 0xFC044004

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	CFBS	
15	14	13	12	11	10	9	8
LOD	–	OPMOD		–	–	SMOD	
7	6	5	4	3	2	1	0
–	–	–	KEYMOD	–	TDESMOD		CIPHER

- **CIPHER: Processing Mode**

0 (DECRYPT): Decrypts data.

1 (ENCRYPT): Encrypts data.

- **TDESMOD: ALGORITHM Mode**

Value	Name	Description
0x0	SINGLE_DES	Single DES processing using TDES_KEY1WRx registers
0x1	TRIPLE_DES	Triple DES processing using TDES_KEY1WRx, TDES_KEY2WRx and TDES_KEY3WRx registers
0x2	XTEA	XTEA processing using TDES_KEY1WRx, TDES_KEY2WRx

Values which are not listed in the table must be considered as “reserved”.

- **KEYMOD: Key Mode**

0: Three-key algorithm is selected.

1: Two-key algorithm is selected. There is no need to write TDES_KEY3WRx registers.

- **SMOD: Start Mode**

Value	Name	Description
0x0	MANUAL_START	Manual Mode
0x1	AUTO_START	Auto Mode
0x2	IDATAR0_START	TDES_IDATAR0 accesses only Auto Mode

Values which are not listed in the table must be considered as “reserved”.

If a DMA transfer is used, 0x2 must be configured. Refer to Section 59.4.3.2 “DMA Mode” for more details.

- **OPMOD: Operating Mode**

Value	Name	Description
0x0	ECB	Electronic Code Book mode
0x1	CBC	Cipher Block Chaining mode
0x2	OFB	Output Feedback mode
0x3	CFB	Cipher Feedback mode

For CBC-MAC operating mode, please set OPMOD to CBC and LOD to 1.

- **LOD: Last Output Data Mode**

0: No effect.

After each end of encryption/decryption, the output data is available either on the output data registers (Manual and Auto modes) .

In Manual and Auto modes, the DATRDY flag is cleared when at least one of the Output Data registers is read.

1: The DATRDY flag is cleared when at least one of the Input Data Registers is written.

No more Output Data Register reads are necessary between consecutive encryptions/decryptions (see Section 59.4.3 “Last Output Data Mode”).

Warning: In DMA mode, reading to the Output Data registers before the last data encryption/decryption process may lead to unpredictable result.

- **CFBS: Cipher Feedback Data Size**

Value	Name	Description
0x0	SIZE_64BIT	64-bit
0x1	SIZE_32BIT	32-bit
0x2	SIZE_16BIT	16-bit
0x3	SIZE_8BIT	8-bit

59.5.3 TDES Interrupt Enable Register

Name: TDES_IER

Address: 0xFC044010

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	URAD
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

- **DATRDY: Data Ready Interrupt Enable**

0: No effect.

1: Enables the corresponding interrupt.

- **URAD: Unspecified Register Access Detection Interrupt Enable**

0: No effect.

1: Enables the corresponding interrupt.

59.5.4 TDES Interrupt Disable Register

Name: TDES_IDR

Address: 0xFC044014

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	URAD
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

- **DATRDY: Data Ready Interrupt Disable**

0: No effect.

1: Disables the corresponding interrupt.

- **URAD: Unspecified Register Access Detection Interrupt Disable**

0: No effect.

1: Disables the corresponding interrupt.

59.5.5 TDES Interrupt Mask Register

Name: TDES_IMR

Address: 0xFC044018

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	URAD
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

- **DATRDY: Data Ready Interrupt Mask**

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

- **URAD: Unspecified Register Access Detection Interrupt Mask**

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

59.5.6 TDES Interrupt Status Register

Name: TDES_ISR

Address: 0xFC04401C

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
		URAT		–	–	–	URAD
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

- **DATRDY: Data Ready (cleared by setting bit START or bit SWRST in TDES_CR or by reading TDES_ODATARx)**

0: Output data is not valid.

1: Encryption or decryption process is completed.

Note: If TDES_MR.LOD = 1: In Manual and Auto modes, the DATRDY flag can also be cleared by writing at least one TDES_IDATARx.

- **URAD: Unspecified Register Access Detection Status (cleared by setting bit TDES_CR.SWRST)**

0: No unspecified register access has been detected since the last write of bit TDES_CR.SWRST.

1: At least one unspecified register access has been detected since the last write of bit TDES_CR.SWRST.

- **URAT: Unspecified Register Access (cleared by setting bit TDES_CR.SWRST)**

Value	Name	Description
0x0	IDR_WR_PROCESSING	Input Data Register written during data processing when SMOD = 0x2 mode.
0x1	ODR_RD_PROCESSING	Output Data Register read during data processing.
0x2	MR_WR_PROCESSING	Mode Register written during data processing.
0x3	WOR_RD_ACCESS	Write-only register read access.

Only the last Unspecified Register Access Type is available through the URAT field.

59.5.7 TDES Key 1 Word Register x

Name: TDES_KEY1WRx

Address: 0xFC044020

Access: Write-only

31	30	29	28	27	26	25	24
KEY1W							
23	22	21	20	19	18	17	16
KEY1W							
15	14	13	12	11	10	9	8
KEY1W							
7	6	5	4	3	2	1	0
KEY1W							

- **KEY1W: Key 1 Word**

The two 32-bit Key 1 Word registers are used to set the 64-bit cryptographic key used for encryption/decryption.

KEY1W0 refers to the first word of the key and KEY1W1 to the last one.

These registers are write-only to prevent the key from being read by another application.

In XTEA mode, the key is defined on 128 bits. These registers contain the 64 LSB bits of the encryption/decryption key.

59.5.8 TDES Key 2 Word Register x

Name: TDES_KEY2WRx

Address: 0xFC044028

Access: Write-only

31	30	29	28	27	26	25	24
KEY2W							

23	22	21	20	19	18	17	16
KEY2W							

15	14	13	12	11	10	9	8
KEY2W							

7	6	5	4	3	2	1	0
KEY2W							

- **KEY2W: Key 2 Word**

The two 32-bit Key 2 Word registers are used to set the 64-bit cryptographic key used for encryption/decryption.

KEY2W0 refers to the first word of the key and KEY2W1 to the last one.

These registers are write-only to prevent the key from being read by another application.

Note: KEY2WRx registers are not used in DES mode.

In XTEA mode, the key is defined on 128 bits. These registers contain the 64 MSB bits of the encryption/decryption key.

59.5.9 TDES Key 3 Word Register x

Name: TDES_KEY3WRx

Address: 0xFC044030

Access: Write-only

31	30	29	28	27	26	25	24
KEY3W							

23	22	21	20	19	18	17	16
KEY3W							

15	14	13	12	11	10	9	8
KEY3W							

7	6	5	4	3	2	1	0
KEY3W							

- **KEY3W: Key 3 Word**

The two 32-bit Key 3 Word registers are used to set the 64-bit cryptographic key used for encryption/decryption.

KEY3W0 refers to the first word of the key and KEY3W1 to the last one.

These registers are write-only to prevent the key from being read by another application.

Note: KEY3WRx registers are not used in DES mode, TDES with two-key algorithm selected and XTEA mode.

59.5.10 TDES Input Data Register x

Name: TDES_IDATARx

Address: 0xFC044040

Access: Write-only

31	30	29	28	27	26	25	24
IDATA							
23	22	21	20	19	18	17	16
IDATA							
15	14	13	12	11	10	9	8
IDATA							
7	6	5	4	3	2	1	0
IDATA							

- **IDATA: Input Data**

The two 32-bit Input Data registers are used to set the 64-bit data block used for encryption/decryption.

IDATA0 refers to the first word of the data to be encrypted/decrypted, and IDATA1 to the last one.

These registers are write-only to prevent the input data from being read by another application.

59.5.11 TDES Output Data Register x

Name: TDES_ODATARx

Address: 0xFC044050

Access: Read-only

31	30	29	28	27	26	25	24
ODATA							
23	22	21	20	19	18	17	16
ODATA							
15	14	13	12	11	10	9	8
ODATA							
7	6	5	4	3	2	1	0
ODATA							

- **ODATA: Output Data**

The two 32-bit Output Data registers contain the 64-bit data block which has been encrypted/decrypted.

ODATA1 refers to the first word, ODATA2 to the last one.

59.5.12 TDES Initialization Vector Register x

Name: TDES_IVRx

Address: 0xFC044060

Access: Write-only

31	30	29	28	27	26	25	24
IV							

23	22	21	20	19	18	17	16
IV							

15	14	13	12	11	10	9	8
IV							

7	6	5	4	3	2	1	0
IV							

- **IV: Initialization Vector**

The two 32-bit Initialization Vector registers are used to set the 64-bit initialization vector data block, which is used by some modes of operation as an additional initial input.

IV1 refers to the first word of the Initialization Vector, IV2 to the last one.

These registers are write-only to prevent the Initialization Vector from being read by another application.

Note: These registers are not used for the ECB mode and must not be written.

59.5.13 TDES XTEA Rounds Register

Name: TDES_XTEA_RNDR

Address: 0xFC044070

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	XTEA_RNDS					

- **XTEA_RNDS: Number of Rounds**

This 6-bit field is used to define the number of complete rounds (1 complete round = 2 Feistel rounds) processed in XTEA algorithm.

The value of XTEA_RNDS has no effect if the TDESMOD field in TDES_MR is set to 0x0 or 0x1.

Note: 0x00 corresponds to 1 complete round, 0x01 corresponds to 2 complete rounds, etc.

60. True Random Number Generator (TRNG)

60.1 Description

The True Random Number Generator (TRNG) passes the American *NIST Special Publication 800-22* (*A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*) and the *Diehard Suite of Tests*.

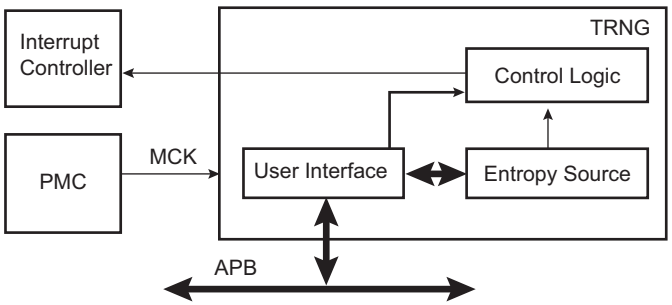
The TRNG may be used as an entropy source for seeding an NIST approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3.

60.2 Embedded Characteristics

- Passes *NIST Special Publication 800-22* Test Suite
- Passes *Diehard Suite of Tests*
- May be used as Entropy Source for seeding an NIST approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3
- Provides a 32-bit Random Number Every 84 Clock Cycles

60.3 Block Diagram

Figure 60-1. TRNG Block Diagram



60.4 Product Dependencies

60.4.1 Power Management

The TRNG interface may be clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the TRNG user interface clock. The user interface clock is independent from any clock that may be used in the entropy source logic circuitry. The source of entropy can be enabled before enabling the user interface clock.

60.4.2 Interrupt Sources

The TRNG interface has an interrupt line connected to the Interrupt Controller. In order to handle interrupts, the Interrupt Controller must be programmed before configuring the TRNG.

Table 60-1. Peripheral IDs

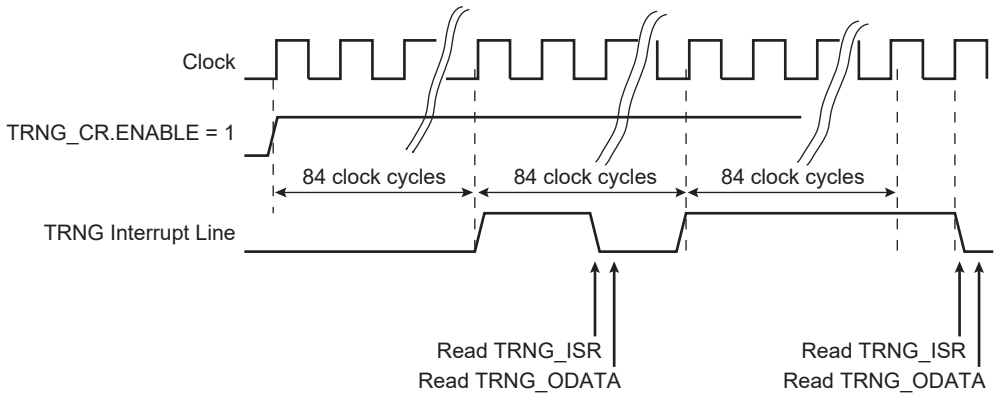
Instance	ID
TRNG	47

60.5 Functional Description

As soon as the TRNG is enabled in the control register (TRNG_CR), the generator provides one 32-bit value every 84 clock cycles. The TRNG interrupt line can be enabled in the TRNG_IER (respectively disabled in the TRNG_IDR). This interrupt is set when a new random value is available and is cleared when the status register (TRNG_ISR) is read. The flag DATRDY of the (TRNG_ISR) is set when the random data is ready to be read out on the 32-bit output data register (TRNG_ODATA).

The normal mode of operation checks that the status register flag equals 1 before reading the output data register when a 32-bit random value is required by the software application.

Figure 60-2. TRNG Data Generation Sequence



60.6 True Random Number Generator (TRNG) User Interface

Table 60-2. Register Mapping

Offset	Register	Name	Access	Reset
0x00	Control Register	TRNG_CR	Write-only	–
0x04–0x0C	Reserved	–	–	–
0x10	Interrupt Enable Register	TRNG_IER	Write-only	–
0x14	Interrupt Disable Register	TRNG_IDR	Write-only	–
0x18	Interrupt Mask Register	TRNG_IMR	Read-only	0x0000_0000
0x1C	Interrupt Status Register	TRNG_ISR	Read-only	0x0000_0000
0x20–0x4C	Reserved	–	–	–
0x50	Output Data Register	TRNG_ODATA	Read-only	0x0000_0000
0x54–0xFC	Reserved	–	–	–

60.6.1 TRNG Control Register

Name: TRNG_CR

Address: 0xFC01C000

Access: Write-only

31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
KEY							
15	14	13	12	11	10	9	8
KEY							
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	ENABLE

- **ENABLE: Enables the TRNG to Provide Random Values**

0: Disables the TRNG.

1: Enables the TRNG if 0x524E47 (“RNG” in ASCII) is written in KEY field at the same time.

- **KEY: Security Key**

Value	Name	Description
0x524E47	PASSWD	Writing any other value in this field aborts the write operation.

60.6.2 TRNG Interrupt Enable Register

Name: TRNG_IER

Address: 0xFC01C010

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

- **DATRDY: Data Ready Interrupt Enable**

0: No effect.

1: Enables the corresponding interrupt.

60.6.3 TRNG Interrupt Disable Register

Name: TRNG_IDR

Address: 0xFC01C014

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

- **DATRDY: Data Ready Interrupt Disable**

0: No effect.

1: Disables the corresponding interrupt.

60.6.4 TRNG Interrupt Mask Register

Name: TRNG_IMR

Address: 0xFC01C018

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

- **DATRDY: Data Ready Interrupt Mask**

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

60.6.5 TRNG Interrupt Status Register

Name: TRNG_ISR

Address: 0xFC01C01C

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
		–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DATRDY

- **DATRDY: Data Ready**

0: Output data is not valid or TRNG is disabled.

1: New random value is completed.

DATRDY is cleared when this register is read.

60.6.6 TRNG Output Data Register

Name: TRNG_ODATA

Address: 0xFC01C050

Access: Read-only

31	30	29	28	27	26	25	24
ODATA							
23	22	21	20	19	18	17	16
ODATA							
15	14	13	12	11	10	9	8
ODATA							
7	6	5	4	3	2	1	0
ODATA							

- **ODATA: Output Data**

The 32-bit Output Data register contains the 32-bit random data.

61. Analog-to-Digital Converter (ADC)

61.1 Description

The ADC is based on a 12-bit Analog-to-Digital Converter (ADC) managed by an ADC Controller providing enhanced resolution up to 14 bits. Refer to Figure 61-1 “Analog-to-Digital Converter Block Diagram”. It also integrates a 12-to-1 analog multiplexer, making possible the analog-to-digital conversions of 12 analog lines. The conversions extend from 0V to the voltage carried on pin ADVREF.

Conversion results are reported in a common register for all channels, as well as in a channel-dedicated register.

The 13-bit and 14-bit resolution modes are obtained by averaging multiple samples to decrease quantization noise. For the 13-bit mode, 4 samples are used, which gives a real sample rate of 1/4 of the actual sample frequency. For the 14-bit mode, 16 samples are used, giving a real sample rate of 1/16 of the actual sample frequency. This arrangement allows conversion speed to be traded off against for better accuracy.

The software trigger, external trigger on rising edge of the ADTRG pin or internal triggers from Timer Counter output(s) are configurable.

The comparison circuitry allows automatic detection of values below a threshold, higher than a threshold, in a given range or outside the range, thresholds and ranges being fully configurable.

The ADC Controller internal fault output is directly connected to the PWM fault input. This input can be asserted by means of comparison circuitry to immediately put the PWM output in a safe state (pure combinational path).

The ADC also integrates a Sleep mode and a conversion sequencer and connects with a DMA channel. These features reduce both power consumption and processor intervention.

This ADC has a selectable single-ended or fully differential input.

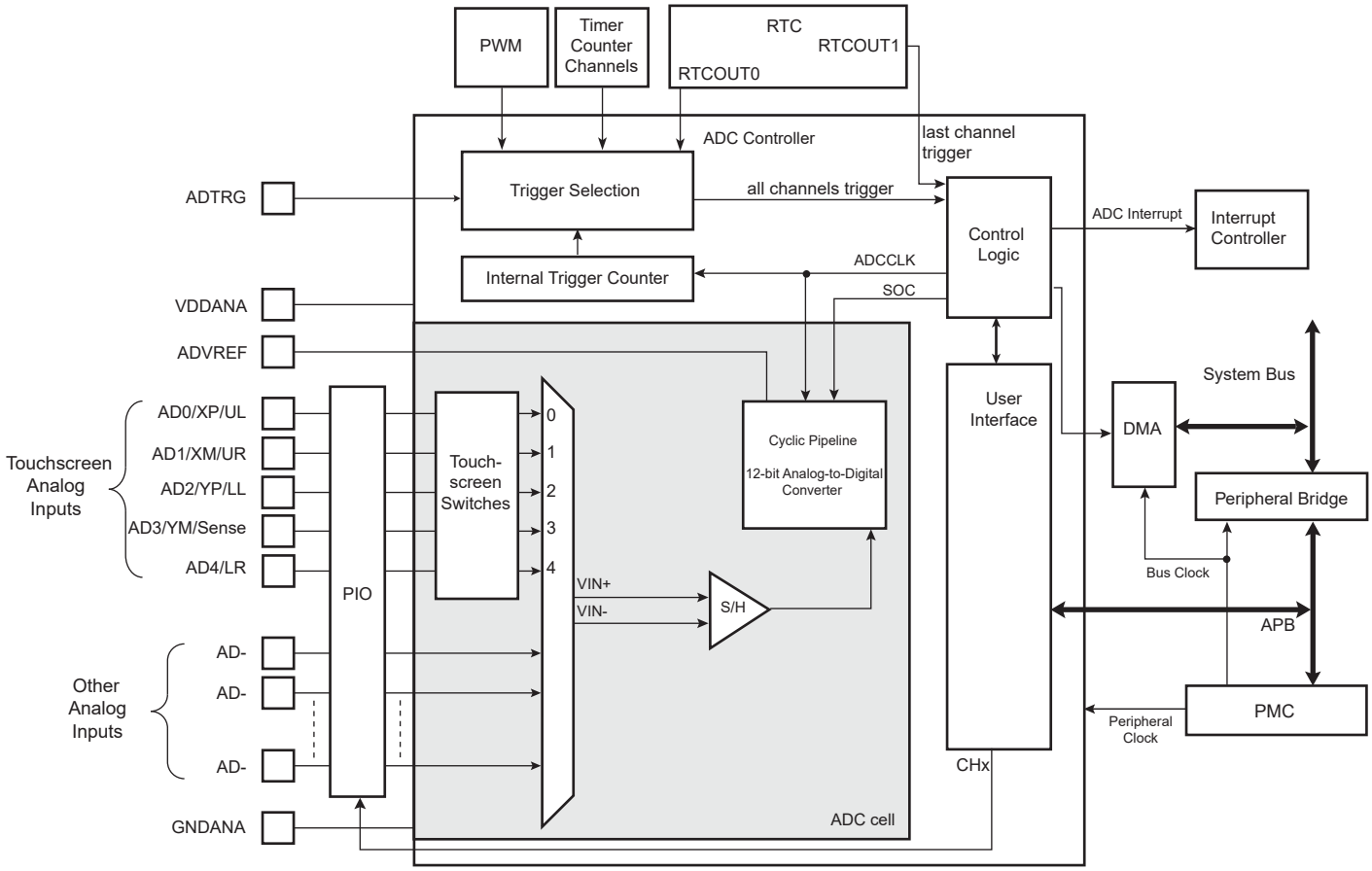
This ADC Controller includes a Resistive Touchscreen Controller. It supports 4-wire and 5-wire technologies.

61.2 Embedded Characteristics

- 12-bit Resolution with Enhanced Mode up to 14 bits
- 1 Msps Conversion Rate
- Digital Averaging Function providing Enhanced Resolution Mode up to 14 bits
- Wide Range of Power Supply Operation
- Selectable Single-Ended or Differential Input Voltage
- Digital correction of offset and gain errors
- Resistive 4-wire and 5-wire Touchscreen Controller
  - Position and Pressure Measurement for 4-wire Screens
  - Position Measurement for 5-wire Screens
  - Average of Up to 8 Measures for Noise Filtering
- Programmable Pen Detection Sensitivity
- Integrated Multiplexer Offering Up to 12 Independent Analog Inputs
- Individual Enable and Disable of Each Channel
- Hardware or Software Trigger from:
  - External Trigger Pin
  - Timer Counter Outputs (Corresponding TIOA Trigger)
  - ADC Internal Trigger Counter
  - Trigger on Pen Contact Detection
  - PWM Event Line
- Drive of PWM Fault Input
- DMA Support
- Two Sleep Modes (Automatic Wakeup on Trigger)
  - Lowest Power Consumption (Voltage Reference OFF Between Conversions)
  - Fast Wakeup Time Response on Trigger Event (Voltage Reference ON Between Conversions)
- Channel Sequence Customization
- Automatic Window Comparison of Converted Values
- Asynchronous Partial Wakeup (SleepWalking) on external trigger
- Register Write Protection

61.3 Block Diagram

Figure 61-1. Analog-to-Digital Converter Block Diagram



61.4 Signal Description

Table 61-1. ADC Pin Description

Pin Name	Description
VDDANA	Analog power supply
ADVREF	Reference voltage
AD0–AD11	Analog input channels
ADTRG	External trigger

61.5 Product Dependencies

61.5.1 Power Management

The ADC Controller is not continuously clocked. The programmer must first enable the ADC Controller peripheral clock in the Power Management Controller (PMC) before using the ADC Controller. However, if the application does not require ADC operations, the ADC Controller clock can be stopped when not needed and restarted when necessary. Configuring the ADC Controller does not require the ADC Controller clock to be enabled.

61.5.2 Interrupt Sources

The ADC interrupt line is connected on one of the internal sources of the Interrupt Controller. Using the ADC interrupt requires the interrupt controller to be programmed first.

Table 61-2. Peripheral IDs

Instance	ID
ADC	40

61.5.3 I/O Lines

The digital input ADTRG is multiplexed with digital functions on the I/O line and the selection of ADTRG is made using the PIO controller.

The analog inputs ADC_ADx are multiplexed with digital functions on the I/O lines. ADC_ADx inputs are selected as inputs of the ADCC when writing a one in the corresponding CHx bit of ADC_CHER and the digital functions are not selected.

Table 61-3. I/O Lines

Instance	Signal	I/O Line	Peripheral
ADC	ADTRG	PD31	A
ADC	AD0	PD19	X1
ADC	AD1	PD20	X1
ADC	AD2	PD21	X1
ADC	AD3	PD22	X1
ADC	AD4	PD23	X1
ADC	AD5	PD24	X1
ADC	AD6	PD25	X1
ADC	AD7	PD26	X1
ADC	AD8	PD27	X1
ADC	AD9	PD28	X1
ADC	AD10	PD29	X1
ADC	AD11	PD30	X1

61.5.4 Hardware Triggers

The ADC can use internal signals to start conversions. Refer to the ADC_MR.TRGSEL field description for exact wiring of internal triggers.

61.5.5 Fault Output

The ADC Controller has the FAULT output connected to the FAULT input of PWM. Refer to Section 61.6.18 “Fault Output” and to section “Pulse Width Modulation Controller (PWM)”.

61.6 Functional Description

61.6.1 Analog-to-Digital Conversion

Once the programmed startup time (ADC_MR.STARTUP) has elapsed, ADC conversions are sequenced by three operating times:

- Tracking time—the time for the ADC to charge its input sampling capacitor to the input voltage. When several channels are converted consecutively, the inherent tracking time is 6 ADC clock cycles. However, the tracking time can be increased using the TRACKTIM field in the Mode Register (ADC_MR).
- ADC inherent conversion time—the time for the ADC to convert the sampled analog voltage. This time is constant and is defined from start of conversion to end of conversion.
- Channel conversion period—the effective time between the end of the current channel conversion and the end of the next channel conversion.

Figure 61-2. Sequence of Consecutive ADC Conversions with TRACKTIM = 0

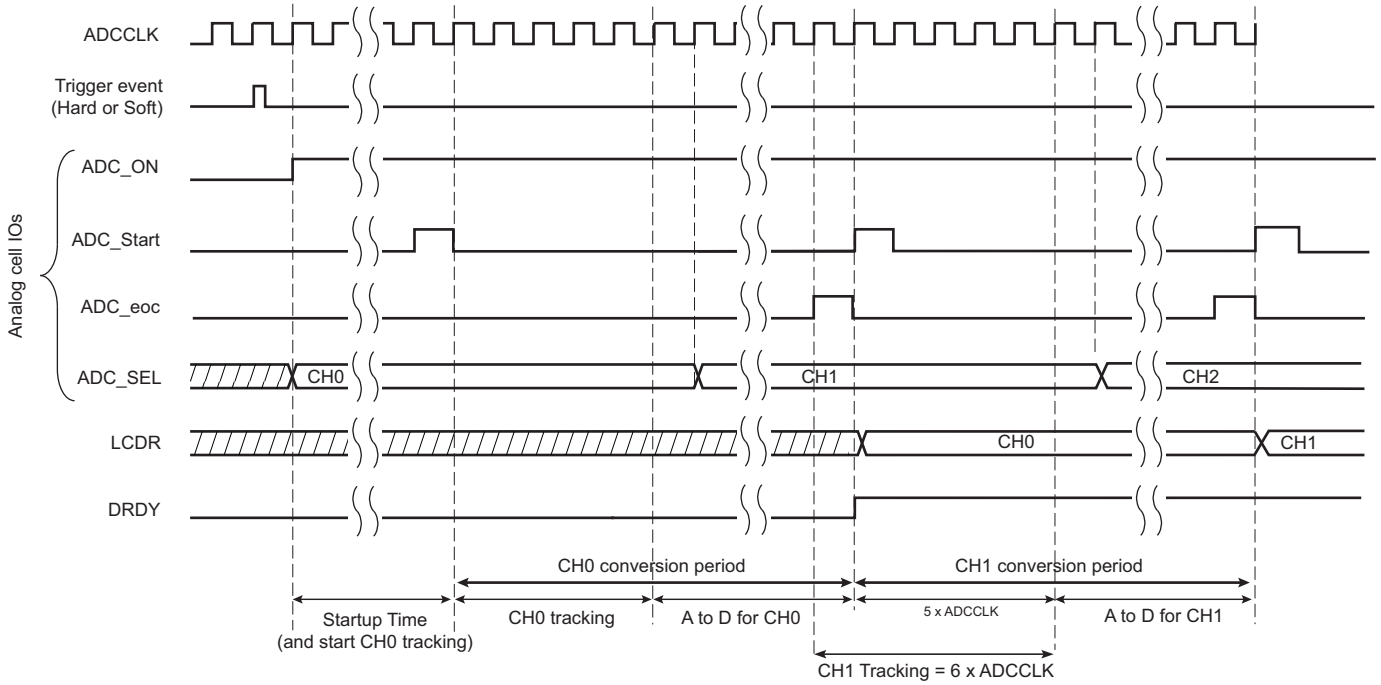
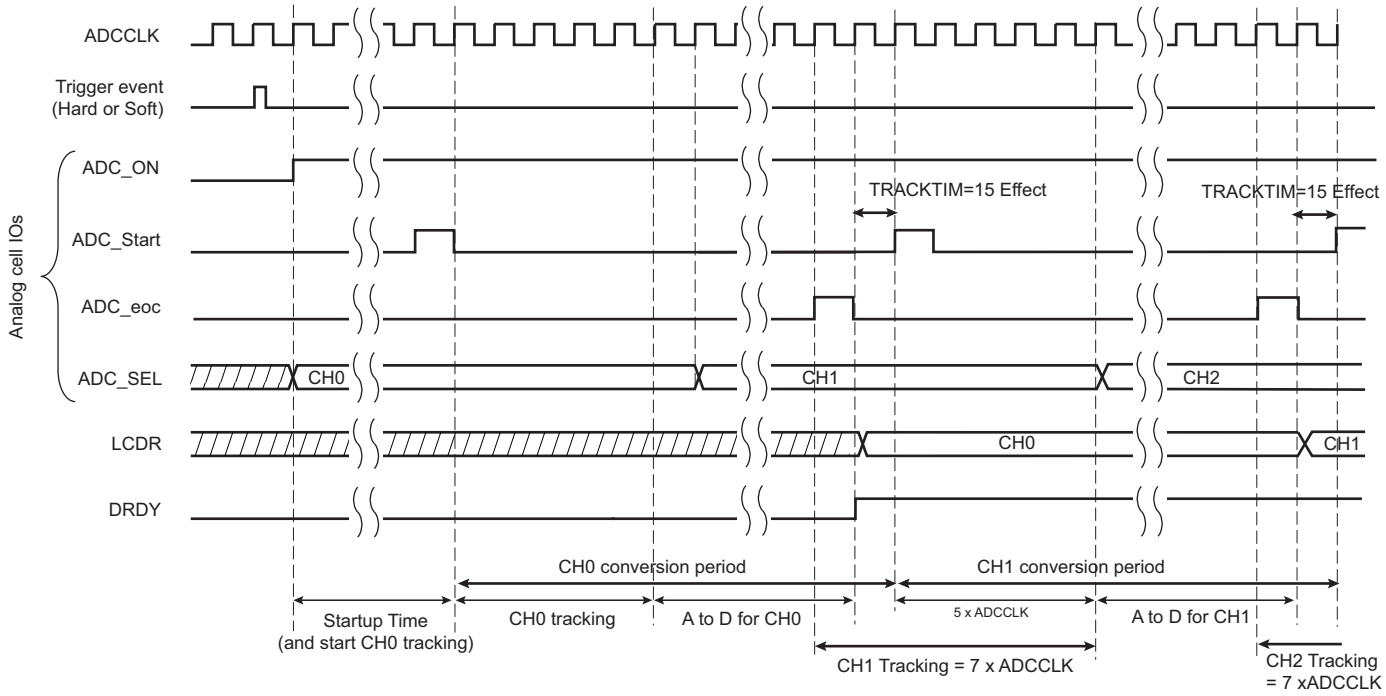


Figure 61-3. Sequence of Consecutive ADC Conversions with TRACKTIM = 15



61.6.2 ADC Clock

The ADC uses the ADC clock (ADCCLK) to perform conversions. The ADC clock frequency is selected in the PRESCAL field of ADC_MR.

To generate the ADC clock, the prescaler has two clock sources: the peripheral clock and the GCLK clock. This clock source is selected using the SRCCLK bit in the Extended Mode Register (ADC_EMR).

If GCLK is selected as a source clock, the ADC clock frequency is independent of the processor/bus clock. At reset, the peripheral clock is selected.

If the SRCCLK bit in ADC_EMR is cleared, the prescaler clock (presc_clk) is driven by peripheral_clock. If the SRCCLK bit in ADC_EMR is set, the prescaler clock is driven by GCLK. The ADC clock frequency is between fpresc_clk/2, if PRESCAL is 0, and fpresc_clk/512, if PRESCAL is set to 255 (0xFF).

PRESCAL must be programmed to provide the ADC clock frequency parameter given in section “Electrical Characteristics”.

61.6.3 ADC Reference Voltage

The voltage reference input of the ADC is the ADVREF pin. Refer to the “Electrical Characteristics” section for further details.

61.6.4 Conversion Resolution

The ADC has a native resolution of 12 bits.

The ADC controller provides enhanced resolution up to 14 bits by means of digital averaging.

If ADTRG is asynchronous to the ADC peripheral clock, the internal resynchronization introduces a jitter of 1 peripheral clock. This jitter may reduce the resolution of the converted signal.

The same applies when using the independent clock (ADC_MR.SRCCLK = 1), if the provided clock is asynchronous to ADC peripheral clock.

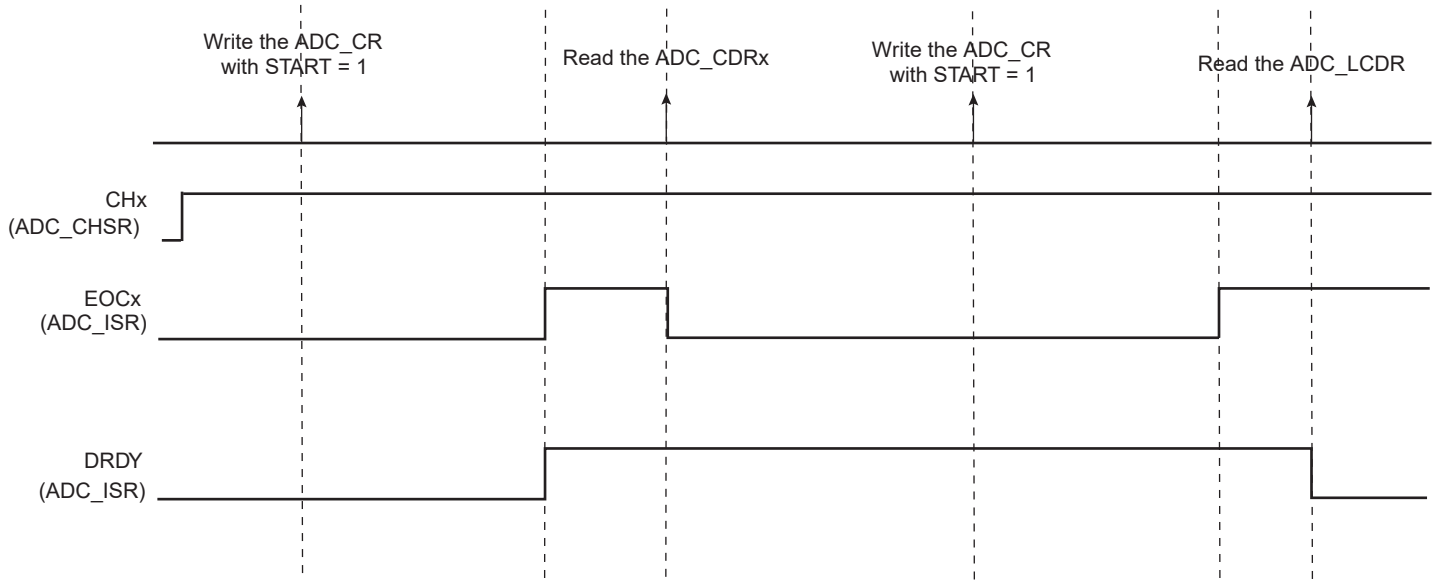
61.6.5 Conversion Results

When a conversion is completed, the resulting digital value is stored in the Channel Data register (ADC_CDRx) of the current channel and in the ADC Last Converted Data register (ADC_LCDR). By setting the TAG option in the Extended Mode Register (ADC_EMR), ADC_LCDR presents the channel number associated with the last converted data in the CHNB field.

When a conversion is completed, the channel EOC bit and the DRDY bit in the Interrupt Status register (ADC_ISR) are set. In the case of a connected DMA channel, DRDY rising triggers a data request. In any case, either EOC and DRDY can trigger an interrupt.

Reading one of the ADC_CDRx clears the corresponding EOC bit. Reading ADC_LCDR clears the DRDY bit.

Figure 61-4. EOCx and DRDY Flag Behavior

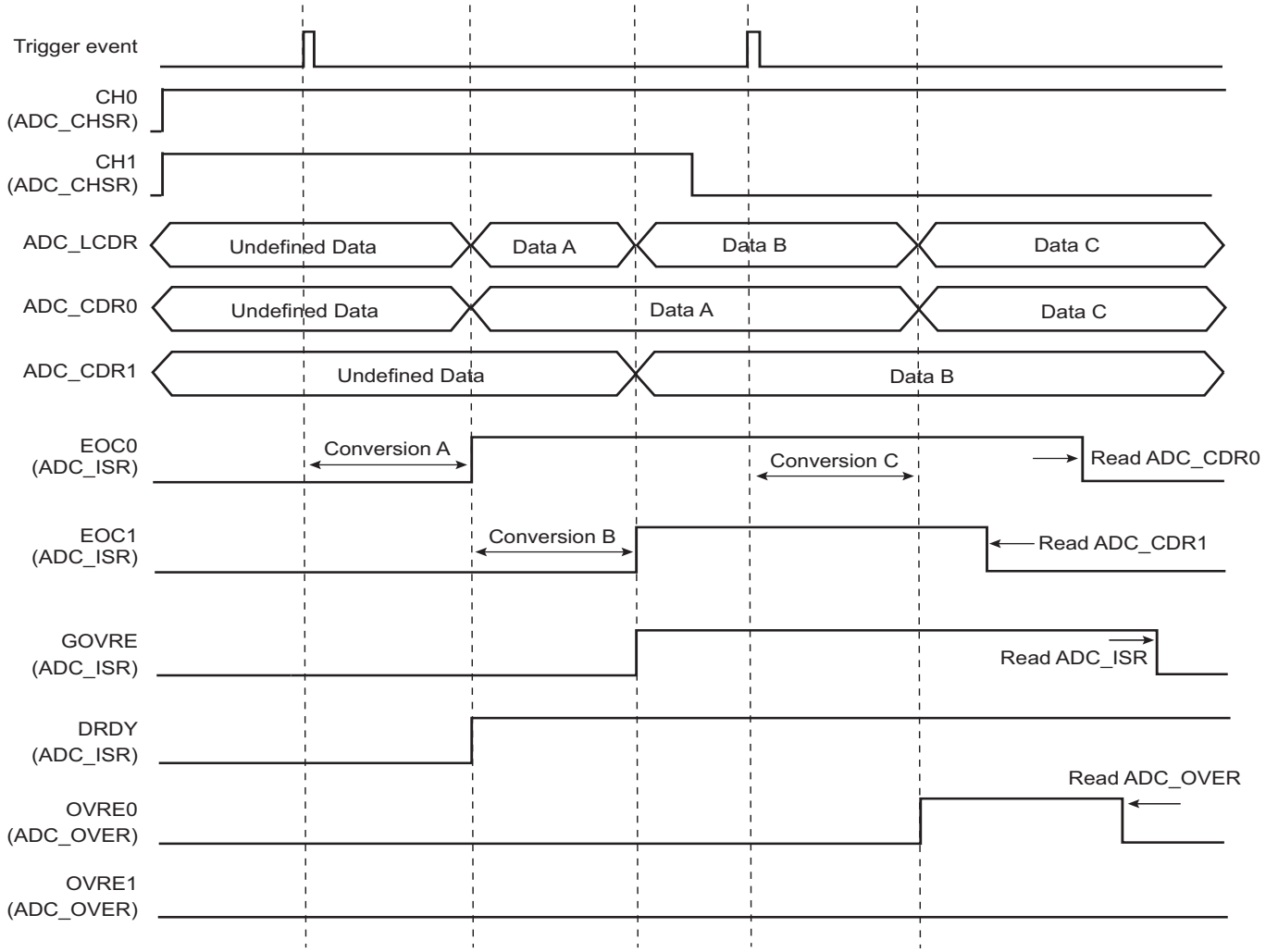


If ADC_CDR is not read before further incoming data is converted, the corresponding OVREx flag is set in the Overrun Status register (ADC_OVER).

If new data is converted when DRDY is high, the GOVRE bit is set in ADC_ISR.

The OVREx flag is automatically cleared when ADC_OVER is read, and the GOVRE flag is automatically cleared when ADC_ISR is read.

Figure 61-5. EOCx, OVREx and GOVREx Flag Behavior



Warning: If the corresponding channel is disabled during a conversion or if it is disabled and then reenabled during a conversion, its associated data and corresponding EOCx and GOVRE flags in ADC_ISR and OVREx flags in ADC_OVER are unpredictable.

61.6.6 Conversion Results Format

The conversion results can be signed (2's complement) or unsigned depending on the value of the SIGNMODE field ("SIGNMODE: Sign Mode" in ADC_EMR).

If conversion results are signed and resolution is less than 16 bits, the sign is extended up to the bit 15 (e.g., 0xF43 for 12-bit resolution is read as 0xFF43, and 0x467 is read as 0x0467).

61.6.7 Conversion Triggers

Conversions of the active analog channels are started with a software or hardware trigger. The software trigger is provided by writing the Control register (ADC_CR) with the START bit at 1.

The list of external/internal events is provided in Section 61.7.2 "ADC Mode Register". The hardware trigger is selected using the TRGSEL field in ADC_MR. The selected hardware trigger is enabled if TRGMOD = 1, 2 or 3 in the ADC Trigger Register (ADC_ TRGR).

The ADC also provides a dual trigger mode (ADC_LCTMR.DUALTRIG = 1) in which the higher index channel can be sampled at a rhythm different from the other channels. The trigger of the last channel is generated by the RTC. Refer to section Section 61.6.12 “Last Channel Specific Measurement Trigger”.

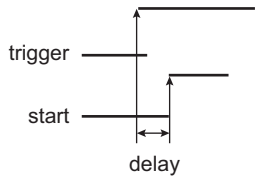
The TRGMOD field in the ADC Trigger register (ADC_TRGR) selects the hardware trigger from the following:

- any edge, either rising or falling or both, detected on the external trigger pin ADTRG
- the Pen Detect, depending on how the PENDET bit is set in the ADC Touchscreen Mode register (ADC_TSMR)
- a continuous trigger, meaning the ADC Controller restarts the next sequence as soon as it finishes the current one
- a periodic trigger, which is defined by programming the TRGPER field in ADC_TRGR

The minimum time between two consecutive trigger events must be strictly greater than the duration time of the longest conversion sequence according to configuration of registers ADC_MR, ADC_CHSR, ADC_SEQRx, and ADC_TSMR.

If a hardware trigger is selected, the start of a conversion is triggered after a delay starting at each rising edge of the selected signal. Due to asynchronous handling, the delay may vary in a range of two peripheral clock periods to one ADC clock period. This delay introduces sampling jitter in the A/D conversion process and may therefore degrade the conversion performance (e.g., SNR, THD).

Figure 61-6. Hardware Trigger Delay



If one of the TIOA outputs is selected, the corresponding Timer Counter channel must be programmed in Waveform mode.

Only one start command is necessary to initiate a conversion sequence on all the channels. The ADC hardware logic automatically performs the conversions on the active channels, then waits for a new request. The Channel Enable (ADC_CHER) and Channel Disable (ADC_CHDR) registers enable the analog channels to be enabled or disabled independently.

If the ADC is used with a DMA, only the transfers of converted data from enabled channels are performed and the resulting data buffers should be interpreted accordingly.

61.6.8 Sleep Mode and Conversion Sequencer

The ADC Sleep mode maximizes power saving by automatically deactivating the ADC when it is not being used for conversions. Sleep mode is selected by setting the SLEEP bit in ADC_MR.

Sleep mode is managed by a conversion sequencer, which automatically processes the conversions of all channels at lowest power consumption.

This mode can be used when the minimum period of time between two successive trigger events is greater than the startup period of the ADC. See section “Electrical Characteristics”.

When a start conversion request occurs, the ADC is automatically activated. As the analog cell requires a startup time, the logic waits during this time and starts the conversion on the enabled channels. When all conversions are complete, the ADC is deactivated until the next trigger. Events triggered during the sequence are ignored.

The conversion sequencer allows automatic processing with minimum processor intervention and optimized power consumption. Conversion sequences can be performed periodically using the internal timer (ADC_TRGR) or the

PWM event line. The periodic acquisition of several samples can be processed automatically without any intervention of the processor via the DMA.

The sequence can be customized by programming the Sequence Channel Registers ADC_SEQR1 and ADC_SEQR2 and setting the USEQ bit of the Mode Register (ADC_MR). The user can choose a specific order of channels and can program up to 12 conversions by sequence. The user is free to create a personal sequence by writing channel numbers in ADC_SEQR1 and ADC_SEQR2. Not only can channel numbers be written in any sequence, channel numbers can be repeated several times. When the bit USEQ in ADC_MR is set, the fields USCHx in ADC_SEQR1 and ADC_SEQR2 are used to define the sequence. Only enabled USCHx fields will be part of the sequence. Each USCHx field has a corresponding enable, CHx-1, in ADC_CHER.

If all ADC channels (i.e., 12) are used on an application board, there is no restriction of usage of the user sequence. However, if some ADC channels are not enabled for conversion but rather used as pure digital inputs, the respective indexes of these channels cannot be used in the user sequence fields (see ADC_SEQRx). For example, if channel 4 is disabled (ADC_CSR[4] = 0), ADC_SEQRx fields USCH1 up to USCH12 must not contain the value 4. Thus the length of the user sequence may be limited by this behavior.

As an example, if only four channels over 12 (CH0 up to CH3) are selected for ADC conversions, the user sequence length cannot exceed four channels. Each trigger event may launch up to four successive conversions of any combination of channels 0 up to 3 but no more (i.e., in this case the sequence CH0, CH0, CH1, CH1, CH1 is impossible).

A sequence that repeats the same channel several times requires more enabled channels than channels actually used for conversion. For example, the sequence CH0, CH0, CH1, CH1 requires four enabled channels (four free channels on application boards) whereas only CH0, CH1 are really converted.

Note: The reference voltage pins always remain connected in Normal mode as in Sleep mode.

61.6.9 Comparison Window

The ADC Controller features automatic comparison functions. It compares converted values to a low threshold, a high threshold or both, depending on the value of the CMPMODE field in ADC_EMR. The comparison can be done on all channels or only on the channel specified in the CMPSEL field of ADC_EMR. To compare all channels, the CMPALL bit of ADC_EMR must be set.

If set, the CMPTYPE bit of ADC_EMR can be used to discard all conversion results that do not match the comparison conditions. Once a conversion result matches the comparison conditions, all the subsequent conversion results are stored in ADC_LCDR (even if these results do not meet the comparison conditions). Setting the CMPRST bit in ADC_CR immediately stops the conversion result storage until the next comparison match.

If the CMPTYPE bit in ADC_EMR is cleared, all conversions are stored in ADC_LCDR. Only the conversions that match the comparison conditions trigger the COMPE flag in ADC_ISR.

Moreover, a filtering option can be set by writing the number of consecutive comparison matches needed to raise the flag. This number can be written and read in the CMPFILTER field of ADC_EMR. The filtering option is dedicated to reinforcing the detection of an analog signal overpassing a predefined threshold. The filter is cleared as soon as ADC_ISR is read, so this filtering function must be used with peripheral DMA controller and works only when using Interrupt mode (no polling).

The flag can be read on the COMPE bit of the Interrupt Status register (ADC_ISR) and can trigger an interrupt.

The high threshold and the low threshold can be read/write in the Compare Window register (ADC_CWR).

Depending on the sign of the conversion, chosen with the SIGNMODE field in the ADC Extended Mode Register, the high threshold and low threshold values must be signed or unsigned to maintain consistency during the comparison. If the conversion is signed, both thresholds must also be signed; if the conversion is unsigned, both thresholds must be unsigned. If comparison occurs on all channels, the SIGNMODE field must be set to ALL_UNSIGNED or ALL_SIGNED and the thresholds must be set accordingly.

61.6.10 Differential and Single-ended Input Modes

61.6.10.1 Input-output Transfer Functions

The ADC can be configured to operate in the following input voltage modes:

- Single-ended—ADC_COR.DIFFx = 0. This is the default mode after a reset.
- Differential—ADC_COR.DIFFx = 1 (see Figure 61-7). In Differential mode, the ADC requires differential input signals having a VDD/2 common mode voltage (refer to the “Electrical Characteristics” section).

The following equations give the unsigned ADC input-output transfer function in each mode(1). With signed conversions (refer to field ADC_EMR.SIGNMODE), subtract 2047 from the ADC_LCDR.DATA value given below.

Single-ended mode:

$$\text{ADC\_LCDR.LDATA} = \frac{ADx - GNDANA}{ADVREF - GNDANA} \times 2^{12}$$

Differential mode:

$$\text{ADC\_LCDR.LDATA} = \left(1 + \frac{ADx - \text{ADx+1}}{ADVREF - GNDANA}\right) \times 2^{11}$$

Note: 1. Equations assume ADC_EMR.OSR = 1

If the ANACH bit is set in ADC_MR, the ADC can manage both differential channels and single-ended channels. If the ANACH bit is cleared, the parameters defined in ADC_COR are applied to all channels.

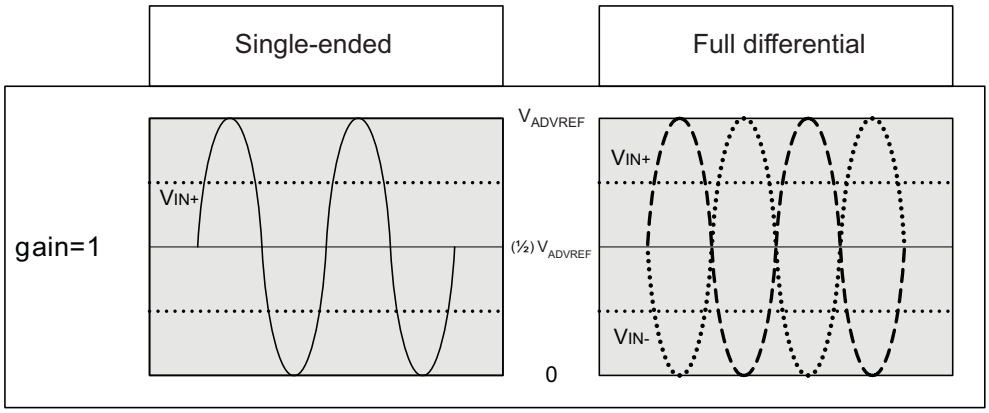
Table 61-4 gives the internal positive and negative ADC inputs assignment with respect to the programmed mode (ADC_COR.DIFFx).

For example, if Differential mode is required on channel 0, input pins AD0 and AD1 are used. In this case, only channel 0 must be enabled by writing a 1 to ADC_CHER.CH0.

Table 61-4. Input Pins and Channel Numbers

Input Pin	Channel Number	
	Single-ended Mode	Differential Mode
AD0	CH0	CH0
AD1	CH1	
AD2	CH2	CH2
AD3	CH3	
AD4	CH4	CH4
AD5	CH5	
AD6	CH6	CH6
AD7	CH7	
AD8	CH8	CH8
AD9	CH9	
AD10	CH10	CH10
AD11	CH11	

Figure 61-7. Analog Full Scale Ranges in Single-Ended/Differential Applications



61.6.11 ADC Timings

The ADC startup time is programmed through the STARTUP field in ADC_MR. See section “Electrical Characteristics”.

The ADC controller provides an inherent tracking time of six ADC clock cycles.

A minimal tracking time is necessary for the ADC to guarantee the best converted final value between two conversions. The tracking time can be adjusted to accommodate a range of source impedances. If more than six ADC clock cycles are required, the tracking time can be increased using the TRACKTIM field in ADC_MR.

Warning: No input buffer amplifier to isolate the source is included in the ADC. See section “Electrical Characteristics”.

61.6.12 Last Channel Specific Measurement Trigger

The last channel (higher index available) embeds a specific mode allowing a measurement trigger period which differs from other active channels. This allows efficient management of the conversions especially if the channel is driven by a device with a variation of a different frequency from other converted channels (for example, but not limited to, temperature sensor).

The last channel can be sampled in different ways through the ADC controller. The different methods of sampling depend on the configuration field TRGMOD in ADC_TRGR and bit CH11 in ADC_CHSR.

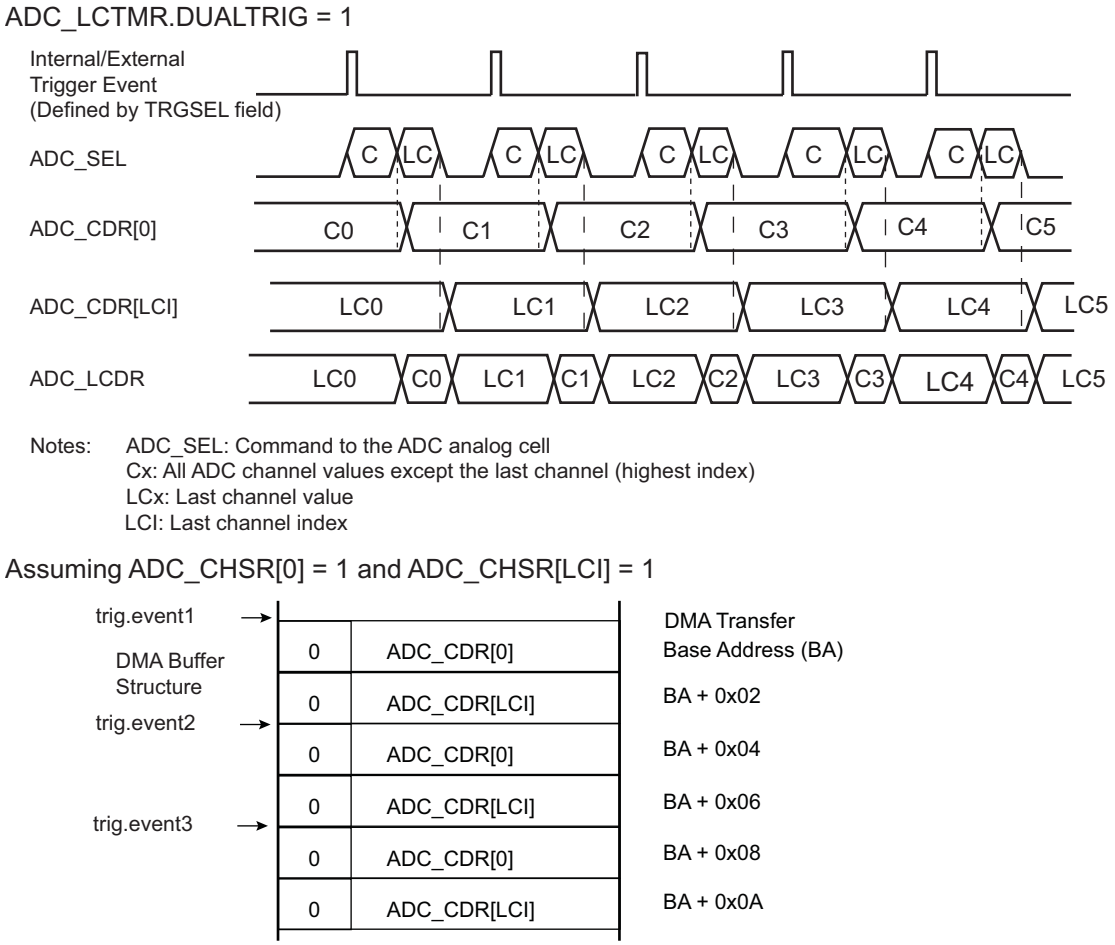
The last channel conversion can be triggered like the other channels by enabling CH11 of ADC_CHER.

The manual start can only be performed if field TRGMOD = 0. When the START bit in ADC_CR is set, the last channel conversion is scheduled together with the other enabled channels (if any). The result of the conversion is placed in ADC_CDR11 register and the associated flag EOC11 is set in ADC_ISR.

If the last channel is enabled in ADC_CHSR, DUALTRIG is cleared and field TRGMOD = 1, 2, 3, 5, the last channel is periodically converted together with the other enabled channels and the result is placed in the ADC_LCDR and ADC_CDR11 registers. Thus the last channel conversion result is part of the DMA Controller buffer (see Figure 61-8).

When the conversion result matches the conditions defined in the ADC_LCTMR and ADC_LCCWR, the LCCHG flag is set in ADC_ISR.

Figure 61-8. Same Trigger for All Channels (ADC_CHSR[LCI] = 1 and ADC_TRGR.TRGMOD = 1, 2, 3, 5)



If the last channel is driven by a device with a slower variation compared to other channels (temperature sensor for example), the channel can be enabled/disabled at any time. However, this may not be optimal for downstream processing.

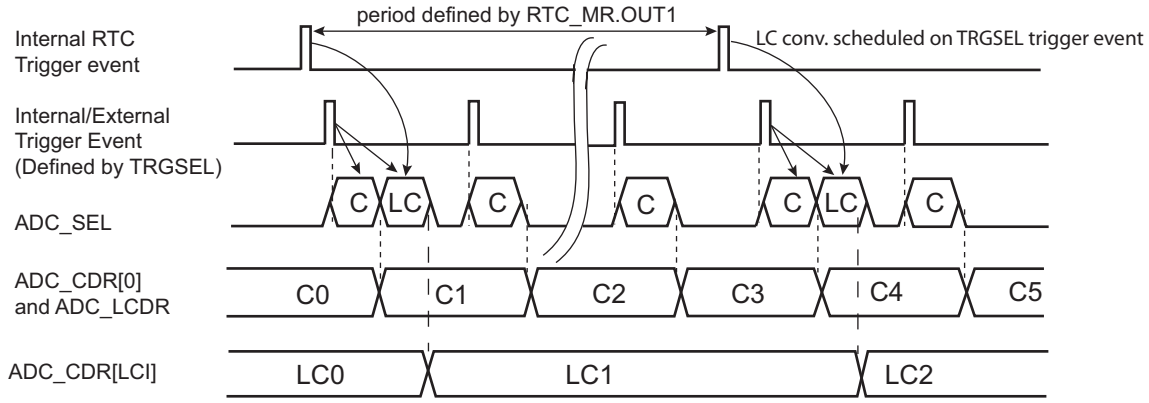
The ADC controller allows a different way of triggering the measurement when DUALTRIG is set in the Last Channel Trigger Mode Register (ADC_LCTMR) but CH11 is not set in ADC_CHSR.

Under these conditions, the last channel conversion is triggered with a period defined by the OUT1 field in the RTC_MR (Real-time Clock Mode Register) while other channels are still active. OUT1 configures an internal trigger generated by the RTC, totally independent of the internal/external triggers. The RTC event will be processed on the next internal/external trigger event as described in Figure 61-9. The internal/external trigger for other channels is selected through the TRGSEL field of ADC_MR.

When DUALTRIG = 1, the result of each conversion of channel 11 is only uploaded in the ADC_CDR11 register and not in ADC_LCDR (see Figure 61-9). Therefore there is no change in the structure of the peripheral DMA controller buffer due to the conversion of the last channel: only the enabled channels are kept in the buffer. The end of conversion of the last channel is reported by the EOC11 flag in ADC_ISR.

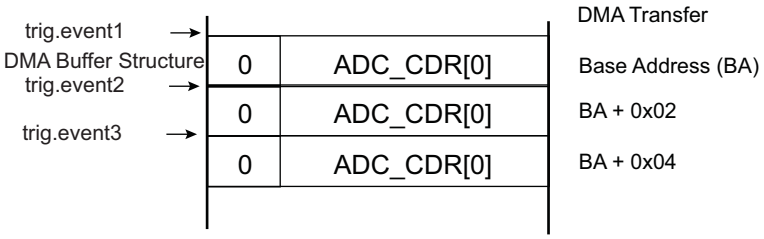
Figure 61-9. Independent Trigger Measurement for Last Channel (ADC_CHSR[LCI] = 0 and ADC_TRGR.TRGMOD = 1, 2, 3, 5)

ADC_LCTMR.DUALTRIG = 1



Notes: ADC_SEL: Command to the ADC analog cell
Cx: All ADC channel values except the last channel (highest index)
LCx: Last channel value
LCI: Last channel index

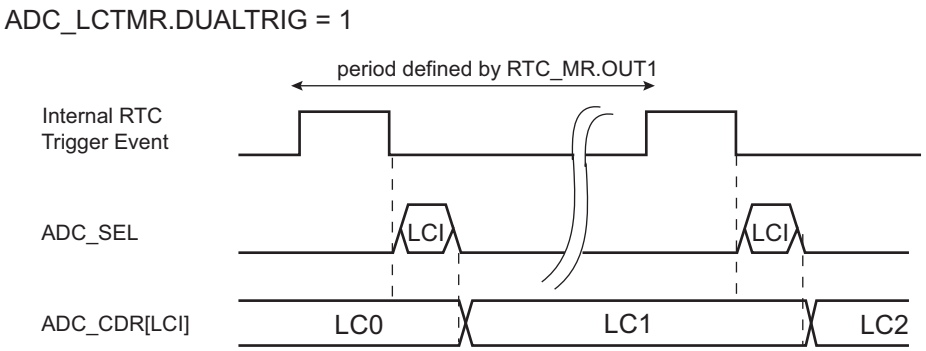
Assuming ADC_CHSR[0] = 1



If DUALTRIG = 1 and field ADC_TRGR.TRGMOD = 0 and none of the channels are enabled in ADC_CHSR (ADC_CHSR = 0), then only channel 11 is converted at a rate defined by the trigger event signal that can be configured in RTC_MR.OUT1 (see Figure 61-10).

This mode of operation, when combined with the Sleep mode operation of the ADC Controller, provides a low-power mode for last channel measure. This assumes there is no other ADC conversion to schedule at a high sampling rate or no other channel to convert.

Figure 61-10. Only Last Channel Measurement Triggered at Low Speed (ADC_CHSR[LCI] = 0 and ADC_TRGR.TRGMOD = 0)



Notes: ADC_SEL: Command to the ADC analog cell
LCx: Last channel value
LCI: Last channel index

61.6.13 Enhanced Resolution Mode and Digital Averaging Function

61.6.13.1 Enhanced Resolution Mode

The Enhanced Resolution mode is enabled if the OSR field is configured to 1 or 2 in ADC_EMR. The enhancement is based on a digital averaging function.

There is no averaging on the last index channel if the measure is triggered by an RTC event.

In this mode, the ADC Controller will trade off conversion speed against accuracy by averaging multiple samples, thus providing a digital low-pass filter function.

The selected oversampling ratio applies to all enabled channels when triggered by an RTC event.

$$\text{ADC\_LCDR.LDATA} = \frac{1}{M} \times \sum_{k=0}^{k=N-1} ADC(k)$$

where N and M are given in the table below.

Table 61-5. Digital Averaging Function Configuration versus OSR Values

ADC_EMR.OSR Value	ADC_LCDR.LDATA Length	N Value	M Value	Full Scale Value	Maximum Value
0	12 bits	1	1	4095	4095
1	13 bits	4	2	8191	8190
2	14 bits	16	4	16383	16381

The average result is valid in ADC_CDRx (x corresponds to the index of the channel) only if the EOCn flag is set in ADC_ISR and if the OVREn flag is cleared in ADC_OVER. The average result for all channels is valid in ADC_LCDR only if DRDY is set and GOVRE is cleared in ADC_ISR.

Note that ADC_CDRs are not buffered. Therefore, when an averaging sequence is ongoing, the value in these registers changes after each averaging sample. However, overrun flags in ADC_OVER rise as soon as the first

sample of an averaging sequence is received. Thus the previous averaged value is not read, even if the new averaged value is not ready.

Consequently, when an overrun flag rises in ADC_OVER, it means that the previous unread data is lost but it does not mean that this data has been overwritten by the new averaged value as the averaging sequence concerning this channel can still be ongoing.

When an oversampling is performed, the maximum value that can be read on ADC_CDRx or ADC_LCDR is not the full-scale value, even if the maximum voltage is supplied on the analog input. Refer to Table 61-5 “Digital Averaging Function Configuration versus OSR Values”.

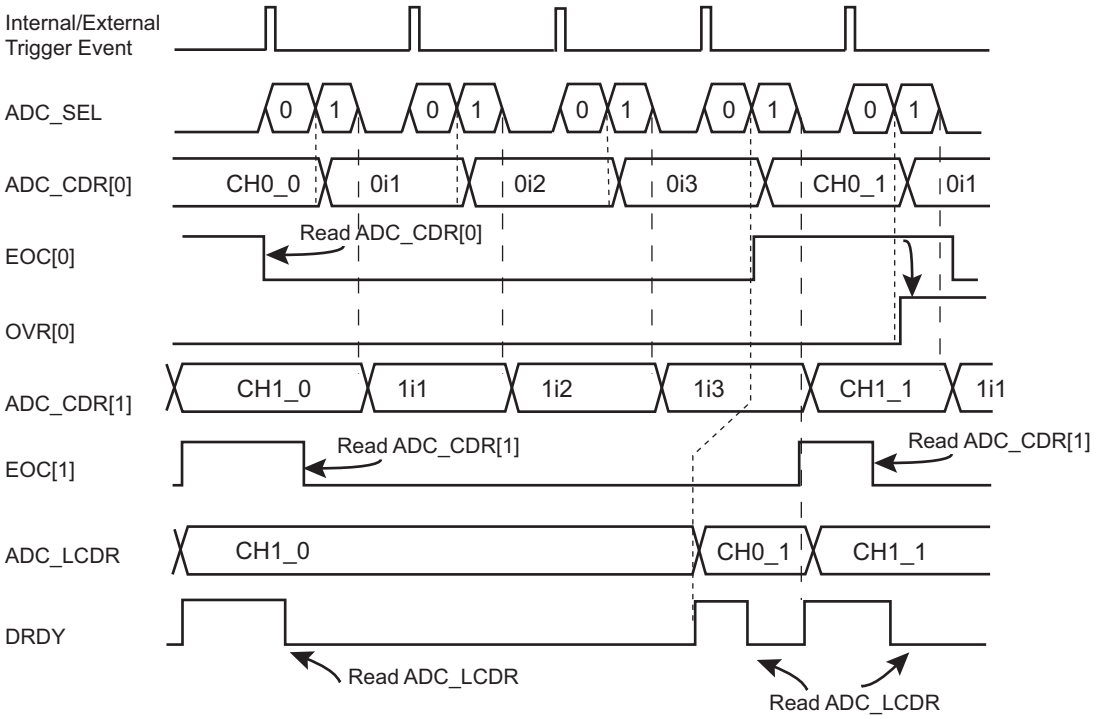
61.6.13.2 Averaging Function versus Trigger Events

The samples can be defined in different ways for the averaging function depending on the configuration of the ASTE bit in ADC_EMR and the USEQ bit in ADC_MR.

When USEQ = 0, there are two possible ways to generate the averaging through the trigger event. If ASTE = 0 in ADC_EMR, every trigger event generates one sample for each enabled channel as described in Figure 61-11. Therefore four trigger events are requested to obtain the result of averaging if OSR = 1.

Figure 61-11. Digital Averaging Function Waveforms Over Multiple Trigger Events

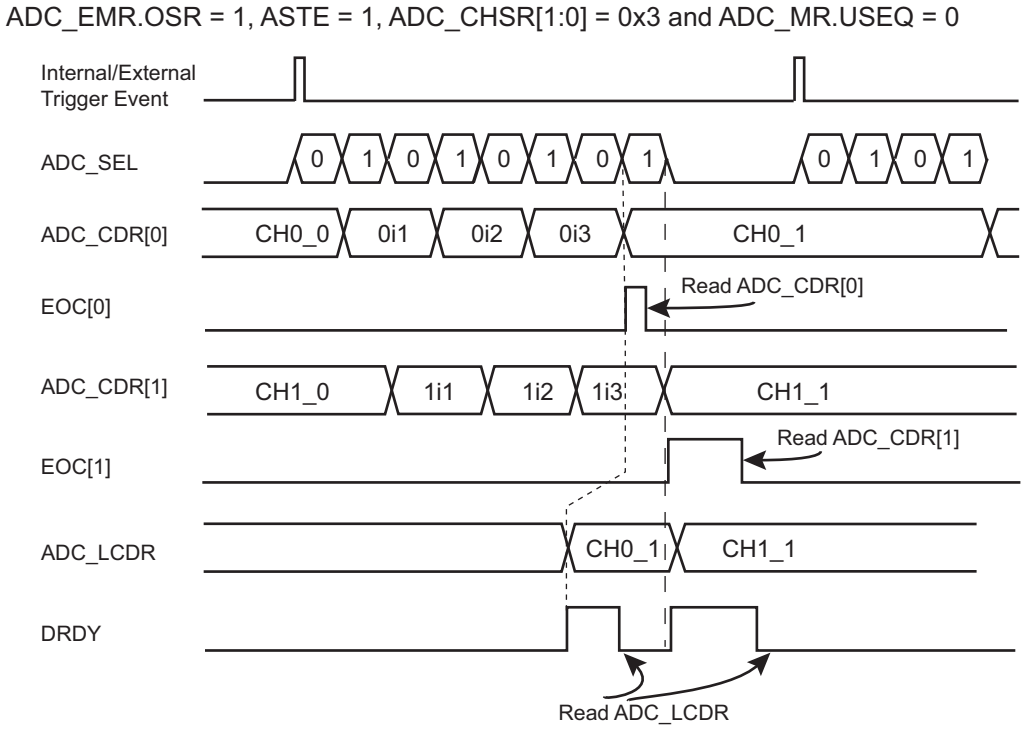
ADC_EMR.OSR = 1, ASTE = 0, ADC_CHSR[1:0] = 0x3 and ADC_MR.USEQ = 0



Note: ADC_SEL: Command to the ADC analog cell
0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0_0, CH0_1, CH1_0 and CH1_1 are final results of average function.

If ASTE = 1 in ADC_EMR and USEQ = 0 in ADC_MR, the sequence to be converted, defined in ADC_CHSR, is automatically repeated n times (where n corresponds to the oversampling ratio defined in the OSR field in ADC_EMR). As a result, only one trigger is required to obtain the result of the averaging function as described in Figure 61-12.

Figure 61-12. Digital Averaging Function Waveforms on a Single Trigger Event



Note: ADC_SEL: Command to the ADC analog cell
0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0_0, CH0_1, CH1_0 and CH1_1 are final results of average function.

When USEQ = 1, the user can define the channel sequence to be converted by configuring ADC_SEQRx and ADC_CHER so that channels are not interleaved during the averaging period. Under these conditions, a sample is defined for each end of conversion as described in Figure 61-13.

When USEQ = 1 and ASTE = 1, OSR can be only configured to 1. Up to three channels can be converted in this mode. The averaging result will be placed in the corresponding ADC_CDRx and in ADC_LCDR for each trigger event. The ADC real sample rate remains the maximum ADC sample rate divided by 4.

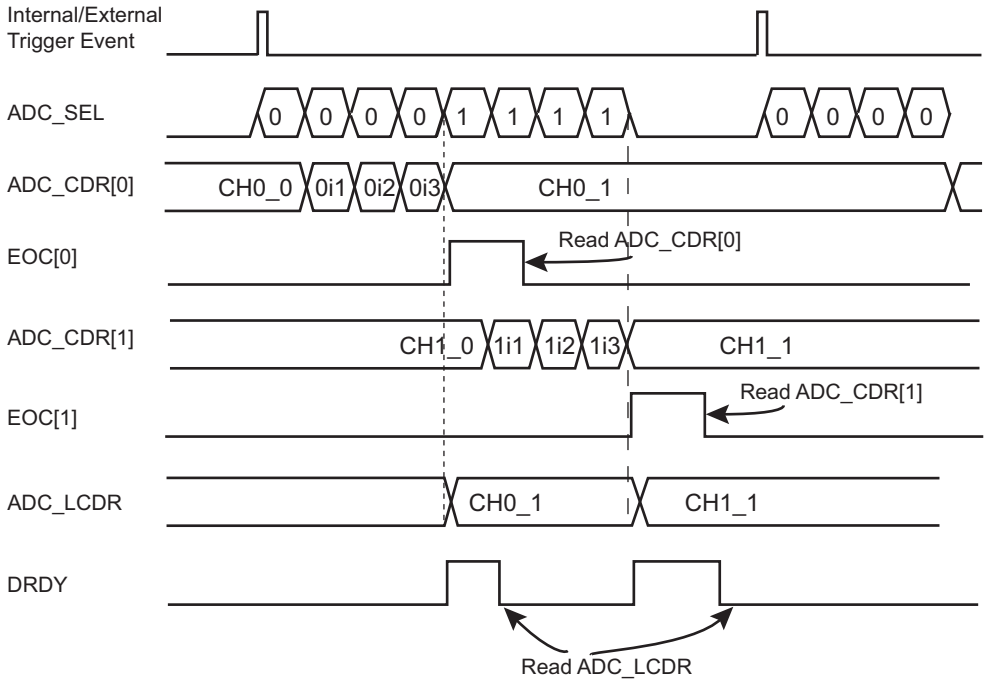
It is important that the user sequence follows a specific pattern. The user sequence must be programmed in such a way that it generates a stream of conversion, where a same channel is successively converted.

Table 61-6. Example Sequence Configurations (USEQ = 1, ASTE = 1, OSR = 1)

	Number of Channels Non-interleaved Averaging - Register Value		
Register	**1 (e.g., CH0)**	**2 (e.g., CH0, CH1)**	**3 (e.g., CH0, CH1, CH2)**
ADC_CHSR	0x0000_000F	0x0000_00FF	0x0000_0FFF
ADC_SEQR1	0x0000_0000	0x1111_0000	0x1111_0000
ADC_SEQR2	0x0000_0000	0x0000_0000	0x0000_2222

Figure 61-13. Digital Averaging Function Waveforms on a Single Trigger Event, Non-interleaved

ADC_EMR.OSR = 1, ASTE = 1, ADC_CHSR[7:0] = 0xFF and ADC_MR.USEQ = 1
ADC_SEQR1 = 0x1111_0000



Note: ADC_SEL: Command to the ADC analog cell
0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0_0, CH0_1, CH1_0 and CH1_1 are final results of average function.

61.6.14 Automatic Error Correction

The ADC features automatic error correction of conversion results. Offset and gain error corrections are available. The correction can be enabled for each channel and correction values (offset and gain) are the same for all channels.

To enable error correction, the corresponding ECORRx bit must be set in the Channel Error Correction Register (ADC_CECR). The offset and gain values used to compensate the results are the same for all correction-enabled channels and programmed in the Correction Values Register (ADC_CVR).

The error correction for channels used with the touchscreen is available in the ADC Touchscreen Correction Values Register (ADC_TSCVR).

The ADCMODE field in ADC_EMR is used to configure a running mode of the ADC Normal mode, Offset Error mode, or Gain Error mode (see Section 61.7.16 “ADC Extended Mode Register”).

After a reset, the running mode of the ADC is Normal mode. Offset Error mode and Gain Error mode are used to determine values of offset compensation and gain compensation, respectively, to apply to conversion results. Table 61-7 provides formulas to obtain the compensation values, with:

- OFFSETCORR—the Offset Correction value. OFFSETCORR is a signed value.
- GAINCORR—the Gain Correction value
- GCi—the intermediate Gain Compensation value
- Gs—the value 13
- ConvValue—the value converted by the ADC (as returned in ADC_LCDR or ADC_CDR)

- Resolution—the resolution used to process the conversion (either RESOLUTION, RESOLUTION+1 or RESOLUTION+2).

Table 61-7. ADC Running Modes

ADC_EMR.ADCMODE	Mode	Description
0	Normal	Normal mode of operation to perform conversions
1	Offset Error	For unsigned conversions: OFFSETCORR = ConvValue – $2^{(\text{Resolution} - 1)}$
		For signed conversions: OFFSETCORR = ConvValue
2	Gain Error	GCi = ConvValue
3		$GAINCORR = \frac{3584}{GCi - ConvValue} \times 2^{(Gs)}$

The final conversion result after error correction is obtained using the following formula:

$$\text{Corrected Data} = (\text{Converted Data} + OFFSETCORR) \times \frac{GAINCORR}{2^{(Gs)}}$$

61.6.15 Touchscreen

61.6.15.1 Touchscreen Mode

The TSMODE parameter of the ADC Touchscreen Mode register (ADC_TSMR) is used to enable/disable the touchscreen functionality, to select the type of screen (4-wire or 5-wire) and, in the case of a 4-wire screen and to activate (or not) the pressure measurement.

In 4-wire mode, channel 0, 1, 2 and 3 must not be used for classic ADC conversions. Likewise, in 5-wire mode, channel 0, 1, 2, 3, and 4 must not be used for classic ADC conversions.

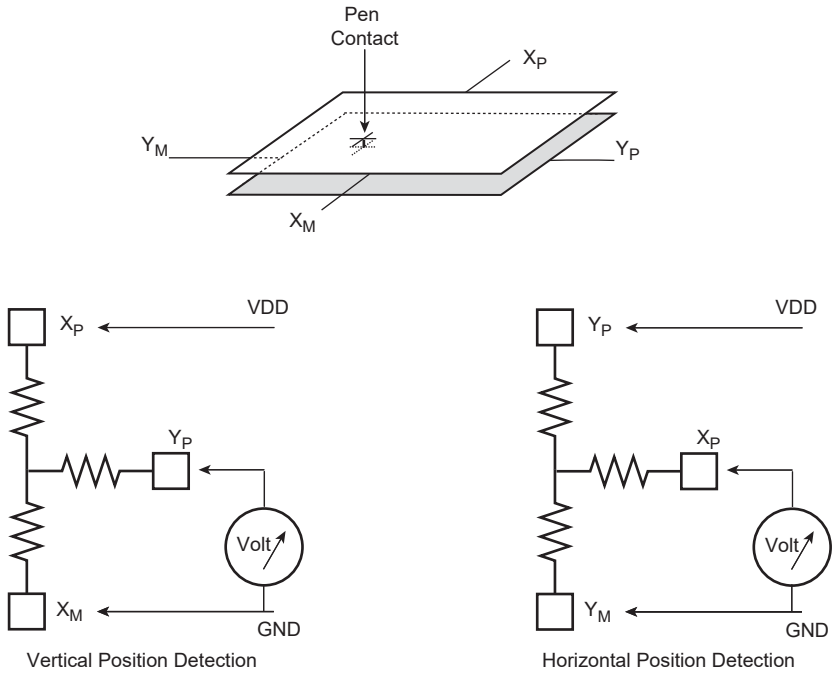
61.6.15.2 4-wire Resistive Touchscreen Principles

A resistive touchscreen is based on two resistive films, each one being fitted with a pair of electrodes, placed at the top and bottom on one film, and on the right and left on the other. In between, there is a layer acting as an insulator, but also enables contact when you press the screen. This is illustrated in Figure 61-14.

The ADC controller can perform the following tasks without external components:

- position measurement
- pressure measurement
- pen detection

Figure 61-14. Touchscreen Position Measurement



61.6.15.3 4-wire Position Measurement Method

As shown in Figure 61-14, to detect the position of a contact, a supply is first applied from top to bottom. Due to the linear resistance of the film, there is a voltage gradient from top to bottom. When a contact is performed on the screen, the voltage propagates at the point the two surfaces come into contact with the second film. If the input impedance on the right and left electrodes sense is high enough, the film does not affect this voltage, despite its resistive nature.

For the horizontal direction, the same method is used, but by applying supply from left to right. The range depends on the supply voltage and on the loss in the switches that connect to the top and bottom electrodes.

In an ideal world (linear, with no loss through switches), the horizontal position is equal to:

VY_M / VDD or VY_P / VDD.

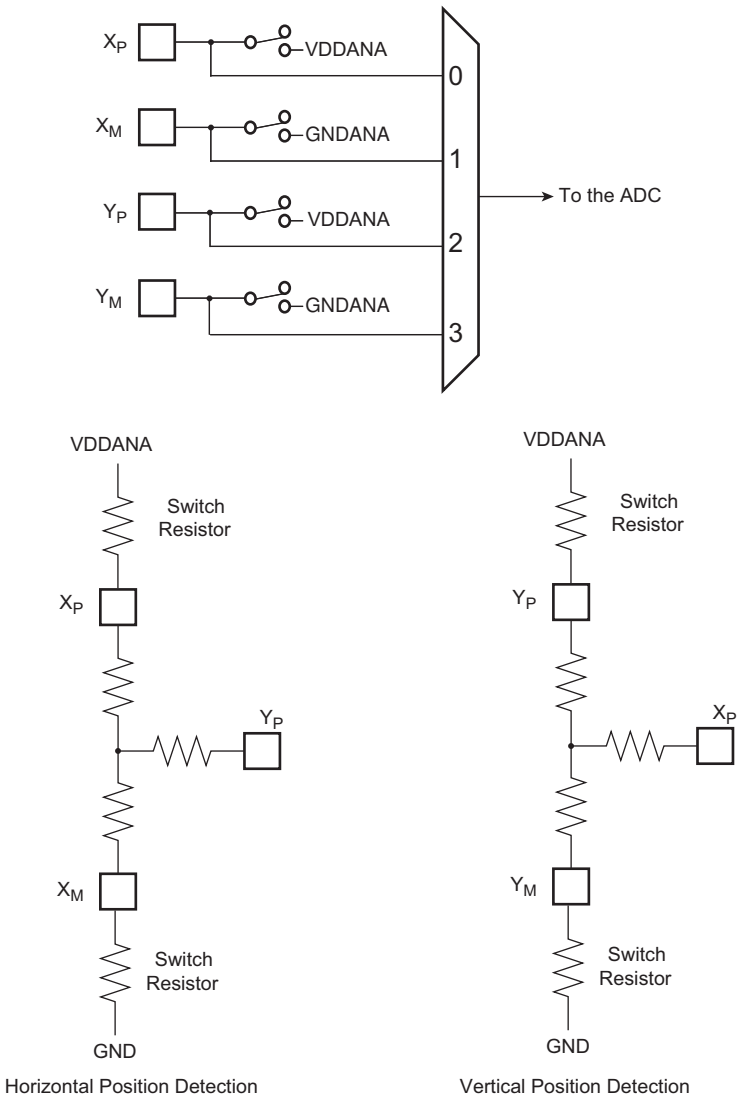
The implementation with on-chip power switches is shown in Figure 61-15. The voltage measurement at the output of the switch compensates for the switches loss.

It is possible to correct for switch loss by performing the operation:

$[VY_P - VX_M] / [VX_P - VX_M]$.

This requires additional measurements, as shown in Figure 61-15.

Figure 61-15. Touchscreen Switches Implementation



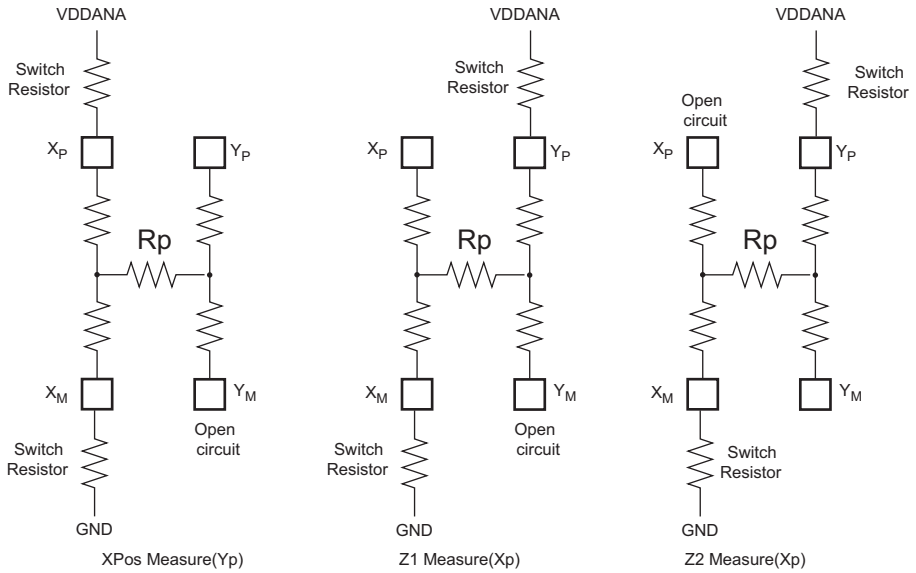
61.6.15.4 4-wire Pressure Measurement Method

The method to measure the pressure (Rp) applied to the touchscreen is based on the known resistance of the X-Panel resistance (Rxp).

Three conversions (Xpos,Z1,Z2) are necessary to determine the value of Rp (Zaxis resistance).

$Rp = Rxp \times (Xpos/1024) \times [(Z2/Z1)-1]$

Figure 61-16. Pressure Measurement



61.6.15.5 5-wire Resistive Touchscreen Principles

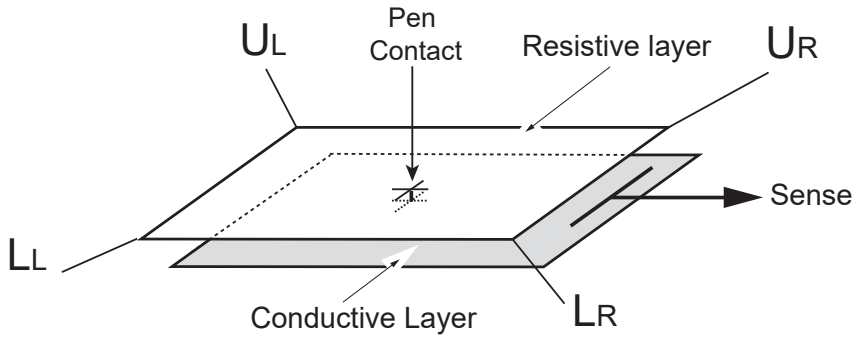
To make a 5-wire touchscreen, a resistive layer with a contact point at each corner and a conductive layer are used.

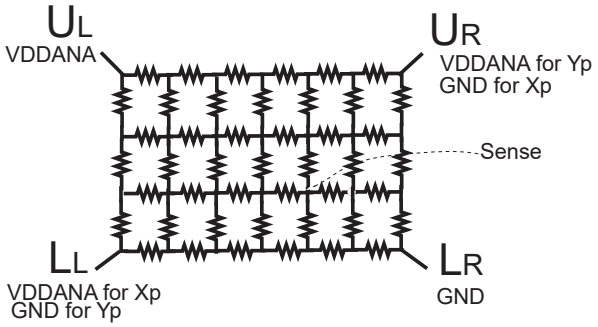
The 5-wire touchscreen differs from the 4-wire type mainly in that the voltage gradient is applied only to one layer, the resistive layer, while the other layer is the sense layer for both measurements.

The measurement of the X position is obtained by biasing the upper left corner and lower left corner to VDDANA and the upper right corner and lower right to ground.

To measure along the Y axis, bias the upper left corner and upper right corner to VDDANA and bias the lower left corner and lower right corner to ground.

Figure 61-17. 5-Wire Principle





61.6.15.6 5-wire Position Measurement Method

In an application only monitoring clicks, 100 points per second is typically needed. For handwriting or motion detection, the number of measurements to consider is approximately 200 points per second. This must take into account that multiple measurements are included (over sampling, filtering) to compute the correct point.

The 5-wire touchscreen panel works by applying a voltage at the corners of the resistive layer and measuring the vertical or horizontal resistive network with the sense input. The ADC converts the voltage measured at the point the panel is touched.

A measurement of the Y position of the pointing device is made by:

- Connecting Upper left (UL) and upper right (UR) corners to VDDANA
- Connecting Lower left (LL) and lower right (LR) corners to ground.

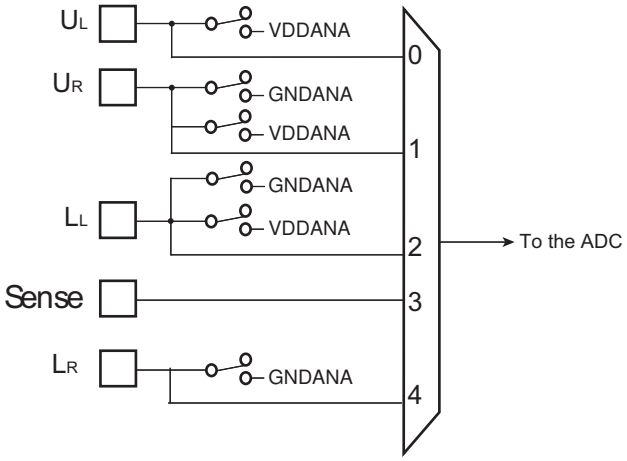
The voltage measured is determined by the voltage divider developed at the point of touch (Y position) and the SENSE input is converted by ADC.

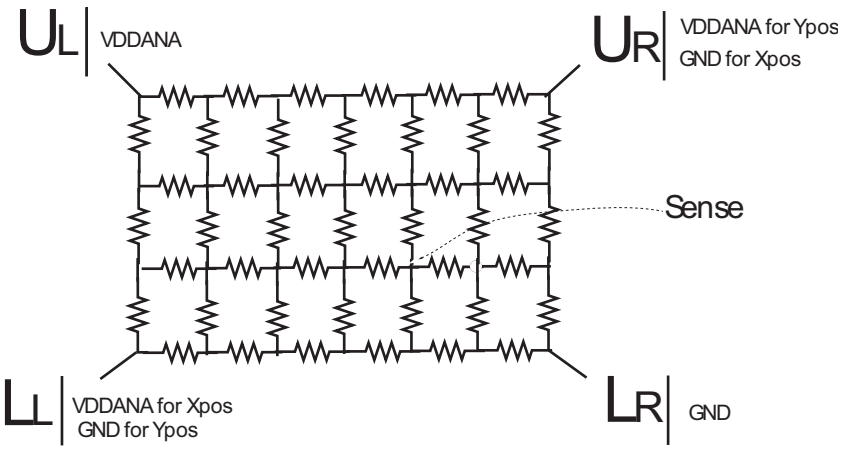
A measurement of the X position of the pointing device is made by:

- Connecting the upper left (UL) and lower left (LL) corners to ground
- Connecting the upper right and lower right corners to VDDANA.

The voltage measured is determined by the voltage divider developed at the point of touch (X position) and the SENSE input is converted by ADC.

Figure 61-18. Touchscreen Switches Implementation



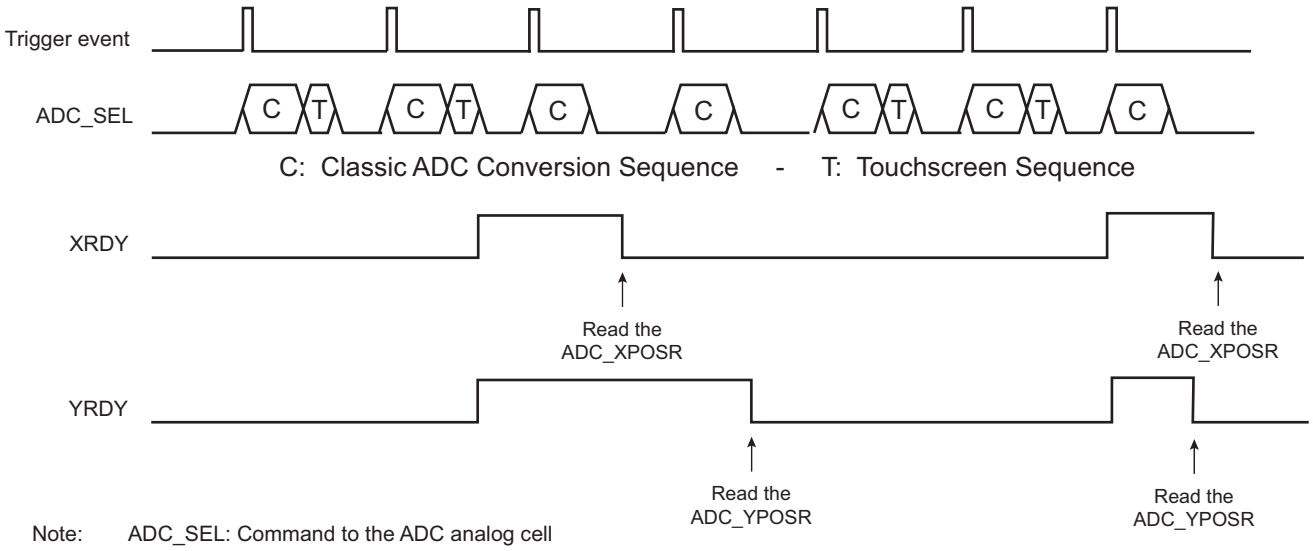


61.6.15.7 Sequence and Noise Filtering

The ADC Controller can manage ADC conversions and touchscreen measurement. On each trigger event the sequence of ADC conversions is performed as described in Section 61.6.8 “Sleep Mode and Conversion Sequencer”. The touchscreen measure frequency can be specified in number of trigger events by writing the TSFREQ parameter in ADC_TSMR. An internal counter counts triggers up to TSFREQ, and every time it rolls out, a touchscreen sequence is appended to the classic ADC conversion sequence (see Figure 61-19).

Additionally the user can average multiple touchscreen measures by writing the TSAV parameter in ADC_TSMR. This can be 1, 2, 4 or 8 measures performed on consecutive triggers as illustrated in Figure 61-19 below. Consequently, the TSFREQ parameter must be greater or equal to the TSAV parameter.

Figure 61-19. Insertion of Touchscreen Sequences (TSFREQ = 2; TSAV = 1)



61.6.15.8 Measured Values, Registers and Flags

As soon as the controller finishes the Touchscreen sequence, XRDY, YRDY and PRDY are set and can generate an interrupt. These flags can be read in the ADC Interrupt Status register (ADC_ISR). They are reset independently by reading in the ADC Touchscreen X Position register (ADC_XPOSR), the ADC Touchscreen Y Position register (ADC_YPOSR) and the ADC Touchscreen Pressure register (ADC_PRESSR).

ADC_XPOSR presents XPOS (VX - VXmin) on its LSB and XSCALE (VXMAX - VXmin) aligned on the 16th bit.

ADC_YPOSR presents YPOS (VY - VYmin) on its LSB and YSCALE (VYMAX - VYmin) aligned on the 16th bit.

To improve the quality of the measure, the user must calculate XPOS/XSCALE and YPOS/YSCALE.

VXMAX, VXmin, VYMAX, and VYmin are measured at the first startup of the controller. These values can change during use, so it can be necessary to refresh them. Refresh can be done by writing ‘1’ in the TSCALIB field of the control register (ADC_CR).

ADC_PRESSR presents Z1 on its LSB and Z2 aligned on the 16th bit. See Section 61.6.15.4 “4-wire Pressure Measurement Method”.

61.6.15.9 Pen Detect Method

When there is no contact, it is not necessary to perform a conversion. However, it is important to detect a contact by keeping the power consumption as low as possible.

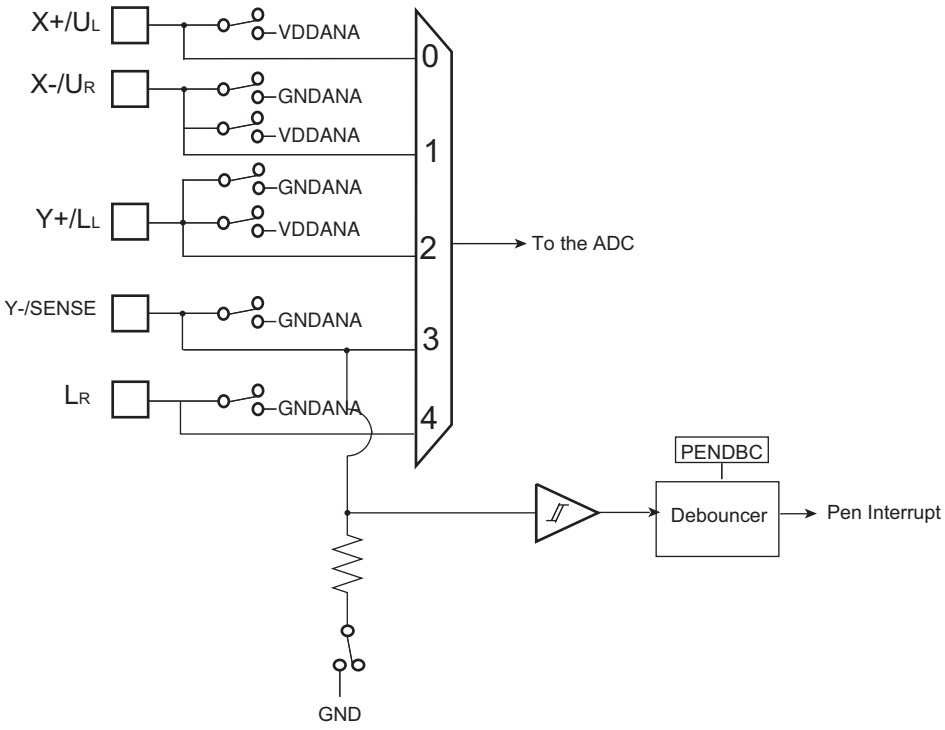
The implementation polarizes one panel by closing the switch on (X_P/U_L) and ties the horizontal panel by an embedded resistor connected to Y_M / Sense. This resistor is enabled by a fifth switch. Since there is no contact, no current is flowing and there is no related power consumption. As soon as a contact occurs, a current is flowing in the Touchscreen and a Schmitt trigger detects the voltage in the resistor.

The Touchscreen Interrupt configuration is entered by programming the PENDET bit in ADC_TSMR. If this bit is written at 1, the controller samples the pen contact state when it is not converting and waiting for a trigger.

To complete the circuit, a programmable debouncer is placed at the output of the Schmitt trigger. This debouncer is programmable up to 2^{15} ADC clock periods. The debouncer length can be selected by programming the field PENDBC in ADC_TSMR.

Due to the analog switch's structure, the debouncer circuitry is only active when no conversion (touchscreen or classic ADC channels) is in progress. Thus, if the time between the end of a conversion sequence and the arrival of the next trigger event is lower than the debouncing time configured on PENDBC, the debouncer will not detect any contact.

Figure 61-20. Touchscreen Pen Detect



The touchscreen pen detect can be used to generate an ADC interrupt to wake up the system. The pen detect generates two types of status, reported in ADC_ISR:

- the PEN bit is set as soon as a contact exceeds the debouncing time as defined by PENDBC and remains set until ADC_ISR is read.
- the NOPEN bit is set as soon as no current flows for a time over the debouncing time as defined by PENDBC and remains set until ADC_ISR is read.

Both bits are automatically cleared as soon as ADC_ISR is read, and can generate an interrupt by writing ADC_IER.

Moreover, the rising of either one of them clears the other, they cannot be set at the same time.

The PENS bit of ADC_ISR shows the current status of the pen contact.

61.6.16 Asynchronous and Partial Wakeup (SleepWalking)

This operating mode is a means of data pre-processing that qualifies an incoming event, thus allowing the ADC to decide whether or not to wake up the system. Asynchronous and partial wakeup is mainly used when the system is in Wait mode (see the PMC section for further details). It can also be enabled when the system is fully running.

Once the Asynchronous and partial wakeup mode is enabled, no access must be performed in the ADC before a wakeup is performed by the ADC.

When the Asynchronous and partial wakeup mode is enabled for the ADC (see the PMC section), the PMC decodes a clock request from the ADC. The clock request is generated as soon as a trigger event occurs. Only a trigger from RTC or ADTRG pin can be used in partial wakeup mode. The selection between RTC or ADTRG pin is performed through the ADC_MR.TRGSEL field.

If the system is in Wait mode (processor and peripheral clocks switched off), the PMC restarts the fast RC oscillator and provides the clock only to the ADC.

To perform a conversion at regular intervals with RTC trigger, the RTC must be configured with the following settings: RTC_MR.OUT0=7 and RTC_MR.THIGH=7. The period of the trigger can be defined in RTC_MR.TPERIOD.

To trigger a conversion using the ADTRG pin, the minimum high level duration of the ADTRG signal must be greater than 2 clock periods of the fast RC oscillator. The maximum duration of the high level must be limited to the amount of startup and conversion time.

As soon as the clock is provided by the PMC, the ADC processes the conversions and compares the converted values with LOWTHRES and HIGHTHRES field values in ADC_CWR.

The ADC instructs the PMC to disable the clock if the converted value does not meet the conditions defined by LOWTHRES and HIGHTHRES field values in ADC_CWR.

If the converted value meets the conditions, the ADC instructs the PMC to exit the full system from Wait mode.

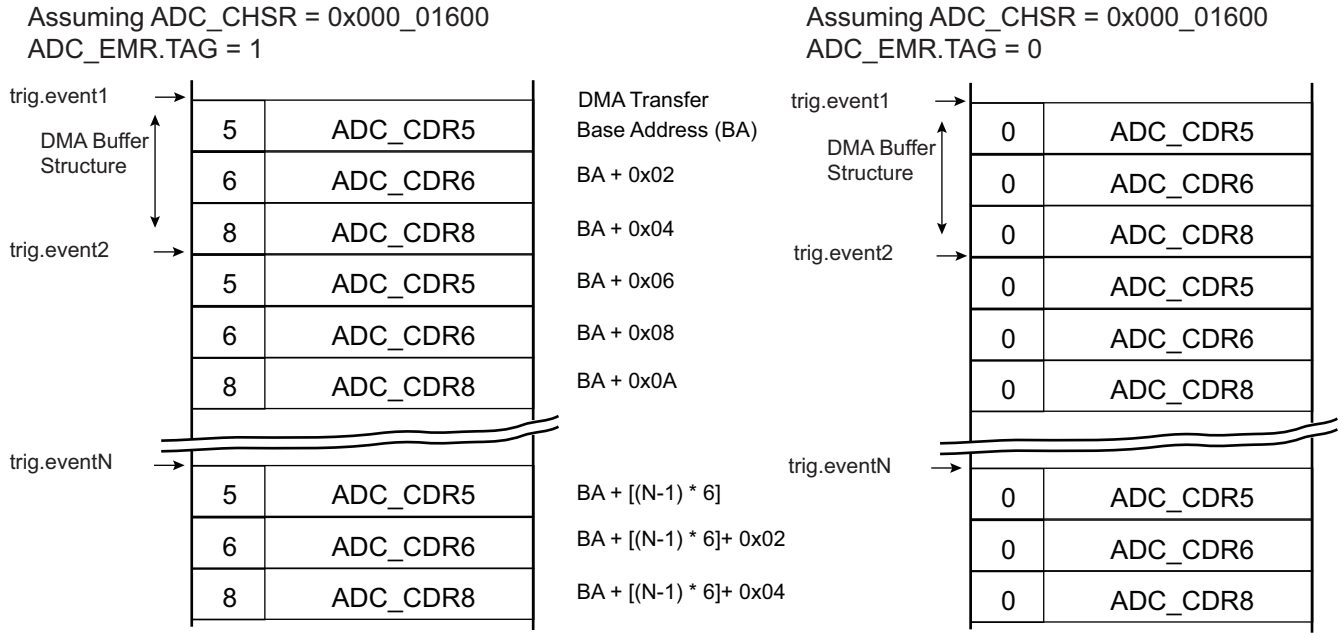
If the processor and peripherals are running, the ADC can be configured in Asynchronous and partial wakeup mode by enabling the PMC_SLPWK_ER (see the PMC section). When a trigger event occurs, the ADC requests the clock from the PMC and the comparison is performed. If there is a comparison match, the ADC continues to request the clock. If there is no match, the clock is switched off for the ADC only, until a new trigger event is detected.

It is recommended to write a ‘1’ to the SLEEP bit to reduce the power consumption of the analog part of the ADC when the system is waiting for a trigger event.

61.6.17 Buffer Structure

The DMA read channel is triggered each time a new data is stored in ADC_LCDR. The same structure of data is repeatedly stored in ADC_LCDR each time a trigger event occurs. Depending on user mode of operation (ADC_MR, ADC_CHSR, ADC_SEQR1, ADC_SEQR2, ADC_TSMR) the structure differs. Each data read to DMA buffer, carried on a half-word (16-bit), consists of last converted data right aligned and when TAG is set in ADC_EMR, the four most significant bits are carrying the channel number thus allowing an easier post-processing in the DMA buffer or better checking the DMA buffer integrity.

Figure 61-21. Buffer Structure



As soon as touchscreen conversions are required, the pen detection function may help the post-processing of the buffer. Refer to Section 61.6.17.4 “Pen Detection Status”.

61.6.17.1 Classic ADC Channels Only (Touchscreen Disabled)

When no touchscreen conversion is required (i.e., TSMODE = 0 in ADC_TSMR), the structure of data within the buffer is defined by ADC_MR, ADC_CHSR, ADC_SEQRx. See Figure 61-21.

If the user sequence is not used (i.e., USEQ is cleared in ADC_MR) then only the value of ADC_CHSR defines the data structure. For each trigger event, enabled channels will be consecutively stored in ADC_LCDR and automatically read to the buffer.

When the user sequence is configured (i.e., USEQ is set in ADC_MR) not only does ADC_CHSR modify the data structure of the buffer, but ADC_SEQRx registers may modify the data structure of the buffer as well.

61.6.17.2 Touchscreen Channels Only

When only touchscreen conversions are required (i.e., TSMODE ≠ 0 in ADC_TSMR and ADC_CHSR equals 0), the structure of data within the buffer is defined by ADC_TSMR.

When TSMODE = 1 or 3, each trigger event adds two half-words in the buffer (assuming TSAV = 0), first half-word being XPOS of ADC_XPOSR then YPOS of ADC_YPOSR. If TSAV/TSFREQ ≠ 0, the data structure remains unchanged. Not all trigger events add data to the buffer.

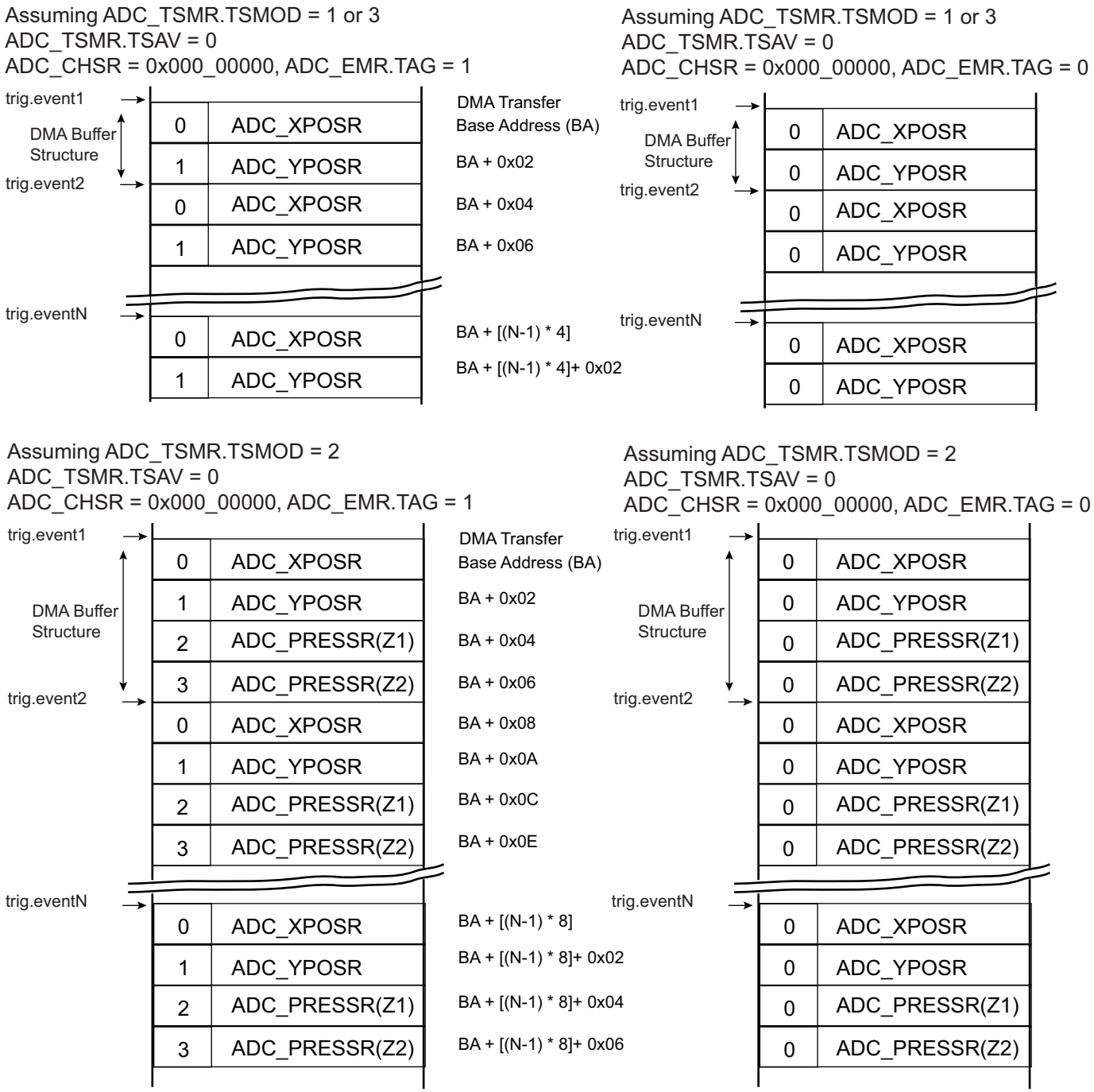
When TSMODE = 2, each trigger event adds four half-words to the buffer (assuming TSAV = 0), first half-word being XPOS of ADC_XPOSR followed by YPOS of ADC_YPOSR and finally Z1 followed by Z2, both located in ADC_PRESSR.

When TAG is set (ADC_EMR), the CHNB field (four most significant bits of ADC_LCDR) is cleared when XPOS is transmitted and set when YPOS is transmitted, allowing an easier post-processing of the buffer or a better checking of the buffer integrity. In case 4-wire with Pressure mode is selected, Z1 value is transmitted to the buffer along with tag set to 2 and Z2 is tagged with value 3.

XSCALE and YSCALE (calibration values) are not transmitted to the buffer because they are supposed to be constant and moreover only measured at the very first startup of the controller or upon user request.

There is no change in buffer structure whatever the value of PENDET bit configuration in ADC_TSMR but it is recommended to use the pen detection function for buffer post-processing (refer to Section 61.6.17.4 “Pen Detection Status”).

Figure 61-22. Buffer Structure When Only Touchscreen Channels are Enabled



61.6.17.3 Interleaved Channels

When both classic ADC channels (CH4/CH5 up to CH12 are set in ADC_CHSR) and touchscreen conversions are required (TSMODE ≠ 0 in ADC_TSMR) the structure of the buffer differs according to TSAV and TSFREQ values.

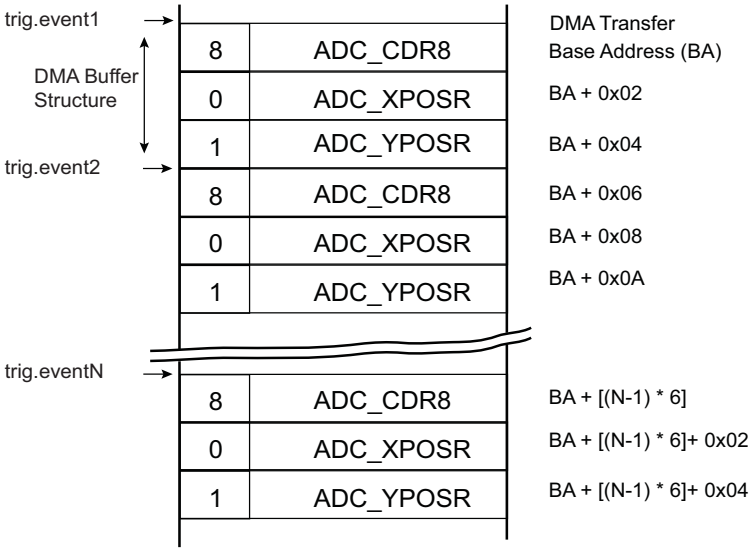
If TSFREQ ≠ 0, not all events generate touchscreen conversions, therefore the buffer structure is based on 2^{TSFREQ} trigger events. Given a TSFREQ value, the location of touchscreen conversion results depends on TSAV value.

When TSFREQ = 0, TSAV must equal 0.

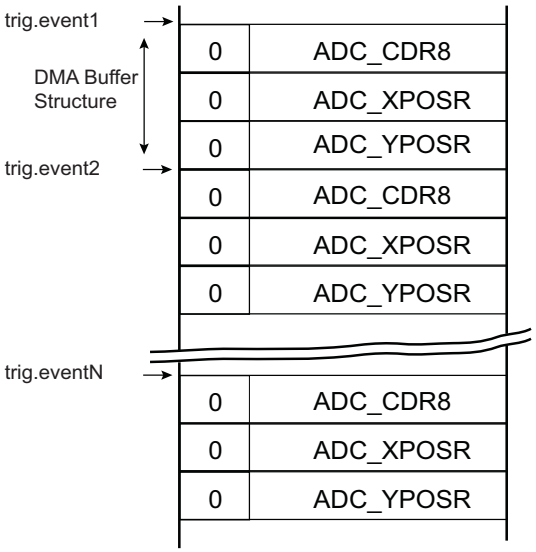
There is no change in buffer structure whatever the value of PENDET bit configuration in ADC_TSMR but it is recommended to use the pen detection function for buffer post-processing (refer to Section 61.6.17.4 “Pen Detection Status”).

Figure 61-23. Buffer Structure When Classic ADC and Touchscreen Channels are Interleaved

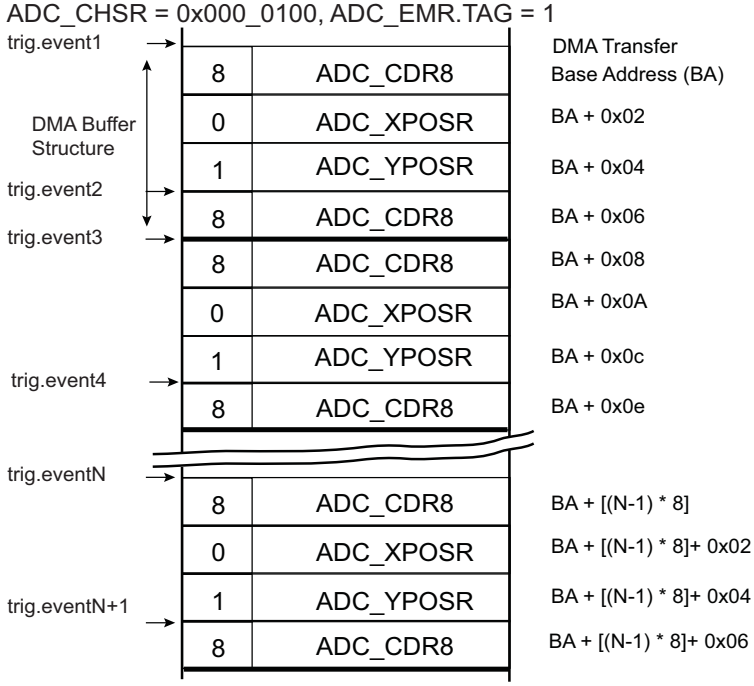
Assuming ADC_TSMR.TSMOD = 1
ADC_TSMR.TSAV = ADC_TSMR.TSFREQ = 0
ADC_CHSR = 0x000_0100, ADC_EMR.TAG = 1



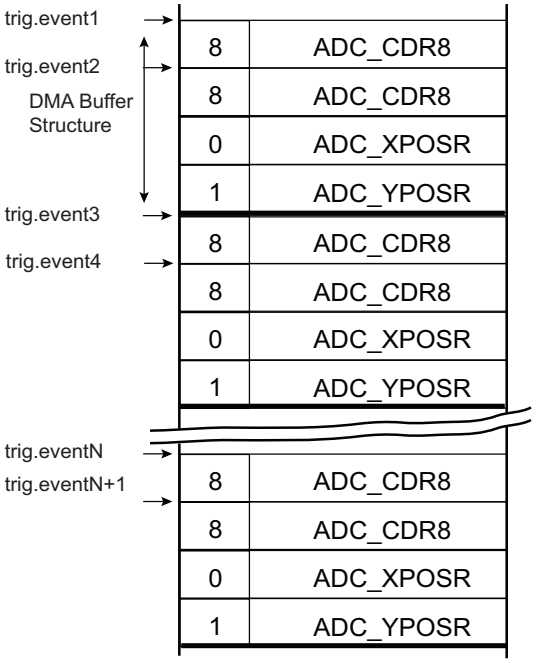
Assuming ADC_TSMR.TSMOD = 1
ADC_TSMR.TSAV = ADC_TSMR.TSFREQ = 0
ADC_CHSR = 0x000_0100, ADC_EMR.TAG = 0



Assuming ADC_TSMR.TSMOD = 1
ADC_TSMR.TSAV = 0, ADC_TSMR.TSFREQ = 1
ADC_CHSR = 0x000_0100, ADC_EMR.TAG = 1



Assuming ADC_TSMR.TSMOD = 1
ADC_TSMR.TSAV = 1, ADC_TSMR.TSFREQ = 1
ADC_CHSR = 0x000_0100, ADC_EMR.TAG = 1



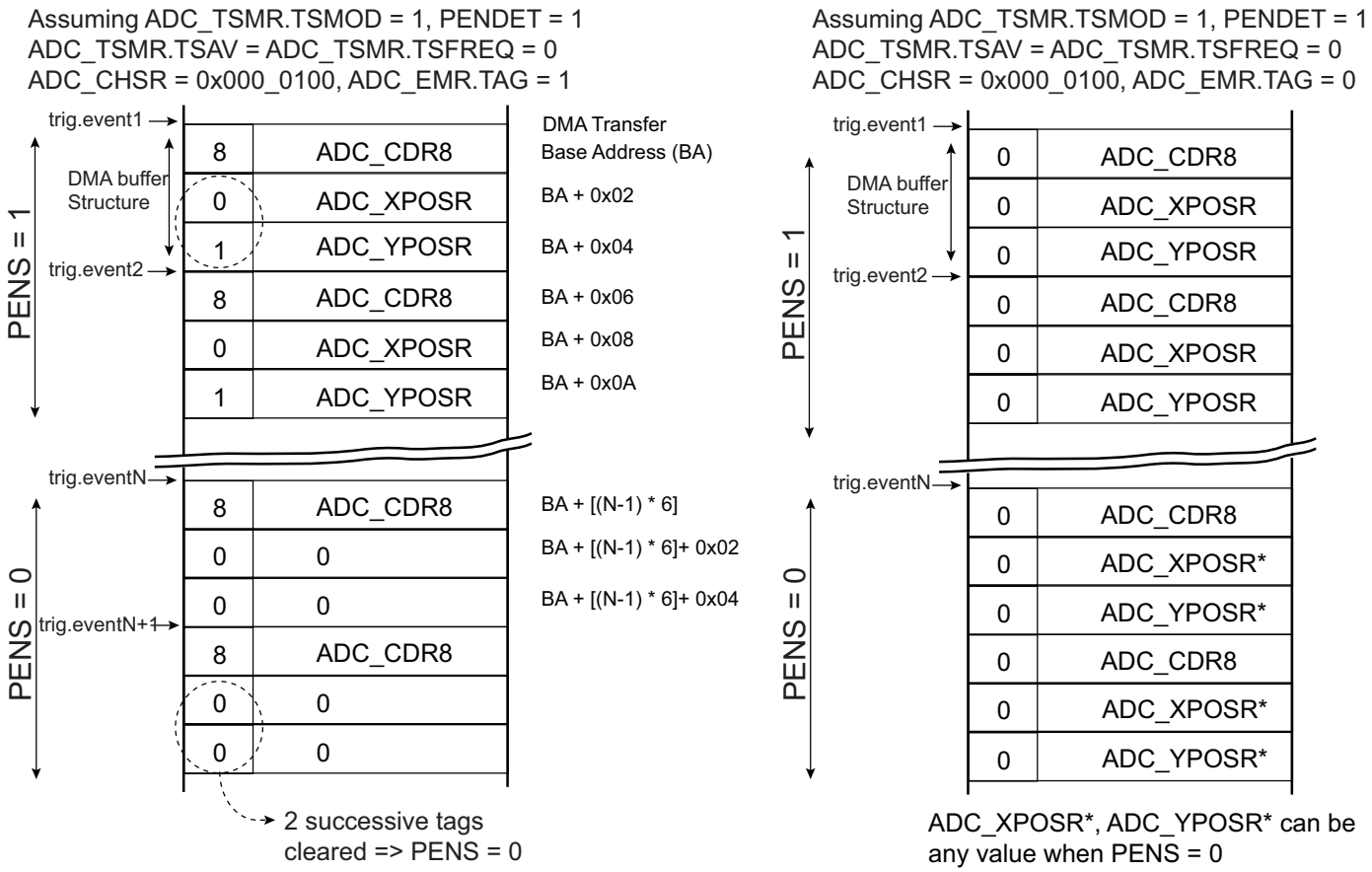
61.6.17.4 Pen Detection Status

If the pen detection measure is enabled (PENDET is set in ADC_TSMR), the XPOS, YPOS, Z1, Z2 values transmitted to the buffer through ADC_LCDR are cleared (including the CHNB field), if the PENS flag of ADC_ISR is 0. When the PENS flag is set, XPOS, YPOS, Z1, Z2 are normally transmitted.

Therefore, using pen detection together with tag function eases the post-processing of the buffer, especially to determine which touchscreen converted values correspond to a period of time when the pen was in contact with the screen.

When the pen detection is disabled or the tag function is disabled, XPOS, YPOS, Z1, Z2 are normally transmitted without tag and no relationship can be found with pen status, thus post-processing may not be easy.

Figure 61-24. Buffer Structure With and Without Pen Detection Enabled



61.6.18 Fault Output

The ADC Controller internal fault output is directly connected to PWM fault input. Fault output may be asserted depending on the configuration of ADC_EMR and ADC_CWR and converted values. When the compare occurs, the ADC fault output generates a pulse of one peripheral clock cycle to the PWM fault input. This fault line can be enabled or disabled within PWM. Should it be activated and asserted by the ADC Controller, the PWM outputs are immediately placed in a safe state (pure combinational path). Note that the ADC fault output connected to the PWM is not the COMPE bit. Thus the Fault mode (FMOD) within the PWM configuration must be FMOD = 1.

61.6.19 Register Write Protection

To prevent any single software error from corrupting ADC behavior, certain registers in the address space can be write-protected by setting the bit WPEN in the "ADC Write Protection Mode Register" (ADC_WPMR).

If a write access to the protected registers is detected, the WPVS flag in the "ADC Write Protection Status Register" (ADC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS flag is automatically reset by reading ADC_WPSR.

The following registers are write-protected when WPEN is set in ADC_WPMR:

- ADC Mode Register
- ADC Channel Sequence 1 Register
- ADC Channel Sequence 2 Register
- ADC Channel Enable Register
- ADC Channel Disable Register
- ADC Last Channel Trigger Mode Register
- ADC Last Channel Compare Window Register
- ADC Extended Mode Register
- ADC Compare Window Register
- ADC Channel Offset Register
- ADC Analog Control Register
- ADC Touchscreen Mode Register
- ADC Trigger Register
- ADC Correction Values Register
- ADC Channel Error Correction Register
- ADC Touchscreen Correction Values Register

61.7 Analog-to-Digital (ADC) User Interface

Table 61-8. Register Mapping

Offset	Register	Name	Access	Reset
0x00	Control Register	ADC_CR	Write-only	–
0x04	Mode Register	ADC_MR	Read/Write	0x00000000
0x08	Channel Sequence Register 1	ADC_SEQR1	Read/Write	0x00000000
0x0C	Channel Sequence Register 2	ADC_SEQR2	Read/Write	0x00000000
0x10	Channel Enable Register	ADC_CHER	Write-only	–
0x14	Channel Disable Register	ADC_CHDR	Write-only	–
0x18	Channel Status Register	ADC_CHSR	Read-only	0x00000000
0x1C	Reserved	–	–	–
0x20	Last Converted Data Register	ADC_LCDR	Read-only	0x00000000
0x24	Interrupt Enable Register	ADC_IER	Write-only	–
0x28	Interrupt Disable Register	ADC_IDR	Write-only	–
0x2C	Interrupt Mask Register	ADC_IMR	Read-only	0x00000000
0x30	Interrupt Status Register	ADC_ISR	Read-only	0x00000000
0x34	Last Channel Trigger Mode Register	ADC_LCTMR	Read/Write	0x00000000
0x38	Last Channel Compare Window Register	ADC_LCCWR	Read/Write	0x00000000
0x3C	Overrun Status Register	ADC_OVER	Read-only	0x00000000
0x40	Extended Mode Register	ADC_EMR	Read/Write	0x00000000
0x44	Compare Window Register	ADC_CWR	Read/Write	0x00000000
0x48	Reserved	–	–	–
0x4C	Channel Offset Register	ADC_COR	Read/Write	0x00000000
0x50	Channel Data Register 0	ADC_CDR0	Read-only	0x00000000
0x54	Channel Data Register 1	ADC_CDR1	Read-only	0x00000000
...	...	...	...	...
0x7C	Channel Data Register 11	ADC_CDR11	Read-only	0x00000000
0x80–0x90	Reserved	–	–	–
0x94	Analog Control Register	ADC_ACR	Read/Write	0x00000101
0x98–0xAC	Reserved	–	–	–
0xB0	Touchscreen Mode Register	ADC_TSMR	Read/Write	0x00000000
0xB4	Touchscreen X Position Register	ADC_XPOSR	Read-only	0x00000000
0xB8	Touchscreen Y Position Register	ADC_YPOSR	Read-only	0x00000000
0xBC	Touchscreen Pressure Register	ADC_PRESSR	Read-only	0x00000000
0xC0	Trigger Register	ADC_TRGR	Read/Write	0x00000000
0xC4–0xD0	Reserved	–	–	–
0xD4	Correction Values Register	ADC_CVR	Read/Write	0x00000000
0xD8	Channel Error Correction Register	ADC_CECR	Read/Write	0x00000000

Table 61-8. Register Mapping (Continued)

Offset	Register	Name	Access	Reset
0xDC	Touchscreen Correction Values Register	ADC_TSCVR	Read/Write	0x00000000
0xE0	Reserved	–	–	–
0xE4	Write Protection Mode Register	ADC_WPMR	Read/Write	0x00000000
0xE8	Write Protection Status Register	ADC_WPSR	Read-only	0x00000000
0xEC–0xFC	Reserved	–	–	–

Note: Any offset not listed in the table must be considered as “reserved”.

61.7.1 ADC Control Register

Name: ADC_CR

Address: 0xFC030000

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	CMPRST	–	TSCALIB	START	SWRST

- **SWRST: Software Reset**

0: No effect.

1: Resets the ADC, simulating a hardware reset.

- **START: Start Conversion**

0: No effect.

1: Begins analog-to-digital conversion.

- **TSCALIB: Touchscreen Calibration**

0: No effect.

1: Programs screen calibration (VDD/GND measurement)

If conversion is in progress, the calibration sequence starts at the beginning of a new conversion sequence. If no conversion is in progress, the calibration sequence starts at the second conversion sequence located after the TSCALIB command (Sleep mode, waiting for a trigger event).

TSCALIB measurement sequence does not affect the Last Converted Data Register (ADC_LCDR).

- **CMPRST: Comparison Restart**

0: No effect.

1: Stops the conversion result storage until the next comparison match.

61.7.2 ADC Mode Register

Name: ADC_MR

Address: 0xFC030004

Access: Read/Write

31	30	29	28	27	26	25	24
USEQ	–	TRANSFER		TRACKTIM			
23	22	21	20	19	18	17	16
ANACH	–	–	–	STARTUP			
15	14	13	12	11	10	9	8
PRESCAL							
7	6	5	4	3	2	1	0
–	FWUP	SLEEP	–	TRGSEL			–

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

- **TRGSEL: Trigger Selection**

Value	Name	Description
0	ADC_TRIG0	ADTRG
1	ADC_TRIG1	TIOA0
2	ADC_TRIG2	TIOA1
3	ADC_TRIG3	TIOA2
4	ADC_TRIG4	PWM event line 0
5	ADC_TRIG5	PWM event line 1
6	ADC_TRIG6	TIOA3
7	ADC_TRIG7	RTCOUT0

Note: The trigger selection can be performed only if TRGMOD = 1, 2 or 3 in ADC Trigger Register.

- **SLEEP: Sleep Mode**

Value	Name	Description
0	NORMAL	Normal Mode: The ADC core and reference voltage circuitry are kept ON between conversions.
1	SLEEP	Sleep Mode: The wakeup time can be modified by programming the FWUP bit.

- **FWUP: Fast Wakeup**

Value	Name	Description
0	OFF	If SLEEP is 1, then both ADC core and reference voltage circuitry are OFF between conversions
1	ON	If SLEEP is 1, then Fast Wakeup Sleep mode: The voltage reference is ON between conversions and ADC core is OFF

- **PRESCAL: Prescaler Rate Selection**

  PRESCAL = $(f_{peripheral\ clock} / (2 \times f_{ADCCLK})) - 1$.

- **STARTUP: Startup Time**

Value	Name	Description
0	SUT0	0 periods of ADCCLK
1	SUT8	8 periods of ADCCLK
2	SUT16	16 periods of ADCCLK
3	SUT24	24 periods of ADCCLK
4	SUT64	64 periods of ADCCLK
5	SUT80	80 periods of ADCCLK
6	SUT96	96 periods of ADCCLK
7	SUT112	112 periods of ADCCLK
8	SUT512	512 periods of ADCCLK
9	SUT576	576 periods of ADCCLK
10	SUT640	640 periods of ADCCLK
11	SUT704	704 periods of ADCCLK
12	SUT768	768 periods of ADCCLK
13	SUT832	832 periods of ADCCLK
14	SUT896	896 periods of ADCCLK
15	SUT960	960 periods of ADCCLK

- **ANACH: Analog Change**

Value	Name	Description
0	NONE	No analog change on channel switching: DIFF0 is used for all channels.
1	ALLOWED	Allows different analog settings for each channel. See ADC Channel Offset Register.

- **TRACKTIM: Tracking Time**

Value	Name	Description
0	ADCCLK6	The tracking time is 6 ADC clock cycles.
1–14	–	The tracking time is 6 ADC clock cycles.
15	ADCCLK7	The tracking time is 7 ADC clock cycles.

- **TRANSFER: Transfer Time**

  The TRANSFER field must be set to 2 to guarantee the optimal transfer time.

- **USEQ: Use Sequence Enable**

Value	Name	Description
0	NUM_ORDER	Normal mode: The controller converts channels in a simple numeric order depending only on the channel index.
1	REG_ORDER	User Sequence mode: The sequence respects what is defined in ADC_SEQR1 and ADC_SEQR2 registers and can be used to convert the same channel several times.

61.7.3 ADC Channel Sequence 1 Register

Name: ADC_SEQR1

Address: 0xFC030008

Access: Read/Write

31	30	29	28	27	26	25	24
USCH8				USCH7			
23	22	21	20	19	18	17	16
USCH6				USCH5			
15	14	13	12	11	10	9	8
USCH4				USCH3			
7	6	5	4	3	2	1	0
USCH2				USCH1			

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

- **USCHx: User Sequence Number x**

The allowed range is 0 up to 11, thus only the sequencer from CH0 to CH11 can be used.

This register activates only if the USEQ field in ADC_MR field is set to ‘1’.

Any USCHx field is processed only if the CHx-1 it in ADC_CHSR reads logical ‘1’, else any value written in USCHx does not add the corresponding channel in the conversion sequence.

Configuring the same value in different fields leads to multiple samples of the same channel during the conversion sequence. This can be done consecutively, or not, according to user needs.

When configuring consecutive fields with the same value, the associated channel is sampled as many time as the number of consecutive values, this part of the conversion sequence being triggered by a unique event.

61.7.4 ADC Channel Sequence 2 Register

Name: ADC_SEQR2

Address: 0xFC03000C

Access: Read/Write

31	30	29	28	27	26	25	24
–				–			
23	22	21	20	19	18	17	16
–				–			
15	14	13	12	11	10	9	8
–				USCH11			
7	6	5	4	3	2	1	0
USCH10				USCH9			

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

- **USCHx: User Sequence Number x**

The sequence number x (USCHx) can be programmed by the Channel number CHy where y is the value written in this field. The allowed range is 0 up to 11. So it is only possible to use the sequencer from CH0 to CH11.

This register activates only if the USEQ field in ADC_MR is set to ‘1’.

Any USCHx field is processed only if the CHx-1 bit in ADC_CHSR reads logical ‘1’. Else, any value written in USCHx does not add the corresponding channel in the conversion sequence.

Configuring the same value in different fields leads to multiple samples of the same channel during the conversion sequence. This can be done consecutively, or not, according to user needs.

61.7.5 ADC Channel Enable Register

Name: ADC_CHER

Address: 0xFC030010

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	CH11	CH10	CH9	CH8
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

- **CHx: Channel x Enable**

0: No effect.

1: Enables the corresponding channel.

Note: If USEQ = 1 in ADC_MR, CHx corresponds to the enable of sequence number x+1 described in ADC_SEQR1 and ADC_SEQR2 (e.g. CH0 enables sequence number USCH1).

61.7.6 ADC Channel Disable Register

Name: ADC_CHDR

Address: 0xFC030014

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	CH11	CH10	CH9	CH8
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

- **CHx: Channel x Disable**

0: No effect.

1: Disables the corresponding channel.

Warning: If the corresponding channel is disabled during a conversion or if it is disabled and then reenabled during a conversion, its associated data and corresponding EOCx and GOVRE flags in ADC_ISR and OVREx flags in ADC_OVER are unpredictable.

61.7.7 ADC Channel Status Register

Name: ADC_CHSR

Address: 0xFC030018

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	CH11	CH10	CH9	CH8
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

- **CHx: Channel x Status**

0: The corresponding channel is disabled.

1: The corresponding channel is enabled.

61.7.8 ADC Last Converted Data Register

Name: ADC_LCDR

Address: 0xFC030020

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	CHNBOSR				
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
LDATA							
7	6	5	4	3	2	1	0
LDATA							

- **LDATA: Last Data Converted**

The analog-to-digital conversion data is placed into this register at the end of a conversion and remains until a new conversion is completed.

If OSR = 0 and TAG = 1 in ADC_EMR, the 4 MSB of LDATA carry the channel number to obtain a packed system memory buffer made of 1 converted data stored in a halfword (16-bit) instead of 1 converted data in a 32-bit word, thus dividing by 2 the size of the memory buffer.

- **CHNBOSR: Channel Number in Oversampling Mode**

Indicates the last converted channel when the TAG bit is set in ADC_EMR and the OSR field is not equal to 0 in ADC_EMR0. If the TAG bit is not set, CHNBOSR = 0.

61.7.9 ADC Interrupt Enable Register

Name: ADC_IER

Address: 0xFC030024

Access: Write-only

31	30	29	28	27	26	25	24
–	NOPEN	PEN	–	–	COMPE	GOVRE	DRDY
23	22	21	20	19	18	17	16
–	PRDY	YRDY	XRDY	LCCHG	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	EOC11	EOC10	EOC9	EOC8
7	6	5	4	3	2	1	0
EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

- **EOCx: End of Conversion Interrupt Enable x**
- **LCCHG: Last Channel Change Interrupt Enable**
- **XRDY: Touchscreen Measure XPOS Ready Interrupt Enable**
- **YRDY: Touchscreen Measure YPOS Ready Interrupt Enable**
- **PRDY: Touchscreen Measure Pressure Ready Interrupt Enable**
- **DRDY: Data Ready Interrupt Enable**
- **GOVRE: General Overrun Error Interrupt Enable**
- **COMPE: Comparison Event Interrupt Enable**
- **PEN: Pen Contact Interrupt Enable**
- **NOPEN: No Pen Contact Interrupt Enable**

61.7.10 ADC Interrupt Disable Register

Name: ADC_IDR

Address: 0xFC030028

Access: Write-only

31	30	29	28	27	26	25	24
–	NOPEN	PEN	–	–	COMPE	GOVRE	DRDY
23	22	21	20	19	18	17	16
–	PRDY	YRDY	XRDY	LCCHG	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	EOC11	EOC10	EOC9	EOC8
7	6	5	4	3	2	1	0
EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

- **EOCx: End of Conversion Interrupt Disable x**
- **LCCHG: Last Channel Change Interrupt Disable**
- **XRDY: Touchscreen Measure XPOS Ready Interrupt Disable**
- **YRDY: Touchscreen Measure YPOS Ready Interrupt Disable**
- **PRDY: Touchscreen Measure Pressure Ready Interrupt Disable**
- **DRDY: Data Ready Interrupt Disable**
- **GOVRE: General Overrun Error Interrupt Disable**
- **COMPE: Comparison Event Interrupt Disable**
- **PEN: Pen Contact Interrupt Disable**
- **NOPEN: No Pen Contact Interrupt Disable**

61.7.11 ADC Interrupt Mask Register

Name: ADC_IMR

Address: 0xFC03002C

Access: Read-only

31	30	29	28	27	26	25	24
–	NOPEN	PEN	–	–	COMPE	GOVRE	DRDY
23	22	21	20	19	18	17	16
–	PRDY	YRDY	XRDY	LCCHG	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	EOC11	EOC10	EOC9	EOC8
7	6	5	4	3	2	1	0
EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

- **EOCx: End of Conversion Interrupt Mask x**
- **LCCHG: Last Channel Change Interrupt Mask**
- **XRDY: Touchscreen Measure XPOS Ready Interrupt Mask**
- **YRDY: Touchscreen Measure YPOS Ready Interrupt Mask**
- **PRDY: Touchscreen Measure Pressure Ready Interrupt Mask**
- **DRDY: Data Ready Interrupt Mask**
- **GOVRE: General Overrun Error Interrupt Mask**
- **COMPE: Comparison Event Interrupt Mask**
- **PEN: Pen Contact Interrupt Mask**
- **NOPEN: No Pen Contact Interrupt Mask**

61.7.12 ADC Interrupt Status Register

Name: ADC_ISR

Address: 0xFC030030

Access: Read-only

31	30	29	28	27	26	25	24
PENS	NOPEN	PEN	–	–	COMPE	GOVRE	DRDY
23	22	21	20	19	18	17	16
–	PRDY	YRDY	XRDY	LCCHG	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	EOC11	EOC10	EOC9	EOC8
7	6	5	4	3	2	1	0
EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0

- **EOCx: End of Conversion x (automatically set / cleared)**

0: The corresponding analog channel is disabled, or the conversion is not finished. This flag is cleared when reading the corresponding ADC_CDRx registers.

1: The corresponding analog channel is enabled and conversion is complete.

- **LCCHG: Last Channel Change (cleared on read)**

0: There is no comparison match (defined in the Last Channel Compare Window register (ADC_LCCWR) since the last read of ADC_ISR.

1: The temperature value reported on ADC_CDR11 has changed since the last read of ADC_ISR, according to what is defined in the Last Channel Trigger Mode register (ADC_LCTMR) and Last Channel Compare Window register (ADC_LCCWR).

- **XRDY: Touchscreen XPOS Measure Ready (cleared on read)**

0: No measure has been performed since the last read of ADC_XPOSR.

1: At least one measure has been performed since the last read of ADC_ISR.

- **YRDY: Touchscreen YPOS Measure Ready (cleared on read)**

0: No measure has been performed since the last read of ADC_YPOSR.

1: At least one measure has been performed since the last read of ADC_ISR.

- **PRDY: Touchscreen Pressure Measure Ready (cleared on read)**

0: No measure has been performed since the last read of ADC_PRESSR.

1: At least one measure has been performed since the last read of ADC_ISR.

- **DRDY: Data Ready (automatically set / cleared)**

0: No data has been converted since the last read of ADC_LCDR.

1: At least one data has been converted and is available in ADC_LCDR.

- **GOVRE: General Overrun Error (cleared on read)**

0: No general overrun error occurred since the last read of ADC_ISR.

1: At least one general overrun error has occurred since the last read of ADC_ISR.

- **COMPE: Comparison Event (cleared on read)**

0: No comparison event since the last read of ADC_ISR.

1: At least one comparison event (defined in ADC_EMR and ADC_CWR) has occurred since the last read of ADC_ISR.

- **PEN: Pen contact (cleared on read)**

0: No pen contact since the last read of ADC_ISR.

1: At least one pen contact since the last read of ADC_ISR.

- **NOPEN: No Pen Contact (cleared on read)**

0: No loss of pen contact since the last read of ADC_ISR.

1: At least one loss of pen contact since the last read of ADC_ISR.

- **PENS: Pen Detect Status**

0: The pen does not press the screen.

1: The pen presses the screen.

Note: PENS is not a source of interruption.

61.7.13 ADC Last Channel Trigger Mode Register

Name: ADC_LCTMR

Address: 0xFC030034

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	CMPMOD		–	–	–	DUALTRIG

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

- **DUALTRIG: Dual Trigger ON**

0: All channels are triggered by event defined by TRGSEL in ADC_MR.

1: Last channel (higher index) trigger period is defined by OUT1 in RTC_MR.

- **CMPMOD: Last Channel Comparison Mode**

Value	Name	Description
0	LOW	Generates the LCCHG flag in ADC_ISR when the converted data is lower than the low threshold of the window.
1	HIGH	Generates the LCCHG flag in ADC_ISR when the converted data is higher than the high threshold of the window.
2	IN	Generates the LCCHG flag in ADC_ISR when the converted data is in the comparison window.
3	OUT	Generates the LCCHG flag in ADC_ISR when the converted data is out of the comparison window.

61.7.14 ADC Last Channel Compare Window Register

Name: ADC_LCCWR

Address: 0xFC030038

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	HIGHTHRES			
23	22	21	20	19	18	17	16
HIGHTHRES							
15	14	13	12	11	10	9	8
–	–	–	–	LOWTHRES			
7	6	5	4	3	2	1	0
LOWTHRES							

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

- **LOWTHRES: Low Threshold**

Low threshold associated to compare settings of ADC_LCTMR.

- **HIGHTHRES: High Threshold**

High threshold associated to compare settings of ADC_LCTMR.

61.7.15 ADC Overrun Status Register

Name: ADC_OVER

Address: 0xFC03003C

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	OVRE11	OVRE10	OVRE9	OVRE8
7	6	5	4	3	2	1	0
OVRE7	OVRE6	OVRE5	OVRE4	OVRE3	OVRE2	OVRE1	OVRE0

- **OVREx: Overrun Error x**

0: No overrun error on the corresponding channel since the last read of ADC_OVER.

1: An overrun error has occurred on the corresponding channel since the last read of ADC_OVER.

61.7.16 ADC Extended Mode Register

Name: ADC_EMR

Address: 0xFC030040

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	ADCMODE		–	SIGNMODE		TAG
23	22	21	20	19	18	17	16
–	–	SRCCLK	ASTE	–	–	OSR	
15	14	13	12	11	10	9	8
–	–	CMPFILTER		–	–	CMPALL	–
7	6	5	4	3	2	1	0
CMPSEL				–	CMPTYPE	CMPMODE	

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

- **CMPMODE: Comparison Mode**

Value	Name	Description
0	LOW	When the converted data is lower than the low threshold of the window, generates the COMPE flag in ADC_ISR or, in Partial Wakeup mode, defines the conditions to exit system from Wait mode.
1	HIGH	When the converted data is higher than the high threshold of the window, generates the COMPE flag in ADC_ISR or, in Partial Wakeup mode, defines the conditions to exit system from Wait mode.
2	IN	When the converted data is in the comparison window, generates the COMPE flag in ADC_ISR or, in Partial Wakeup mode, defines the conditions to exit system from Wait mode.
3	OUT	When the converted data is out of the comparison window, generates the COMPE flag in ADC_ISR or, in Partial Wakeup mode, defines the conditions to exit system from Wait mode.

- **CMPTYPE: Comparison Type**

Value	Name	Description
0	FLAG_ONLY	Any conversion is performed and comparison function drives the COMPE flag.
1	START_CONDITION	Comparison conditions must be met to start the storage of all conversions until the CMPRST bit is set.

- **CMPSEL: Comparison Selected Channel**

If CMPALL = 0: CMPSEL indicates which channel has to be compared.

If CMPALL = 1: No effect.

- **CMPALL: Compare All Channels**

0: Only channel indicated in CMPSEL field is compared.

1: All channels are compared.

- **CMPFILTER: Compare Event Filtering**

Number of consecutive compare events necessary to raise the flag = CMPFILTER+1

When programmed to 0, the flag rises as soon as an event occurs.

Refer to Section 61.6.9 “Comparison Window” when using filtering option (CMPFILTER > 0).

- **OSR: Over Sampling Rate**

Value	Name	Description
0	NO_AVERAGE	No averaging. ADC sample rate is maximum.
1	OSR4	1-bit enhanced resolution by averaging. ADC sample rate divided by 4.
2	OSR16	2-bit enhanced resolution by averaging. ADC sample rate divided by 16.

- **ASTE: Averaging on Single Trigger Event**

Value	Name	Description
0	MULTI_TRIG_AVERAGE	The average requests several trigger events.
1	SINGLE_TRIG_AVERAGE	The average requests only one trigger event.

- **SRCCLK: External Clock Selection**

0 (PERIPH_CLK): The peripheral clock is the source for the ADC prescaler.

1 (GCLK): GCLK is the source clock for the ADC prescaler, thus the ADC clock can be independent of the core/peripheral clock.

- **TAG: Tag of ADC_LCDR**

0: Sets CHNB field to zero in ADC_LCDR.

1: Appends the channel number to the conversion result in ADC_LCDR.

- **SIGNMODE: Sign Mode**

Value	Name	Description
0	SE_UNSG_DF_SIGN	Single-Ended channels: Unsigned conversions. Differential channels: Signed conversions.
1	SE_SIGN_DF_UNSG	Single-Ended channels: Signed conversions. Differential channels: Unsigned conversions.
2	ALL_UNSIGNED	All channels: Unsigned conversions.
3	ALL_SIGNED	All channels: Signed conversions.

If conversion results are signed and resolution is below 16 bits, the sign is extended up to the bit 15 (for example, 0xF43 for 12-bit resolution will be read as 0xFF43 and 0x467 will be read as 0x0467). See Section 61.6.6 “Conversion Results Format”.

- **ADCMODE: ADC Running Mode**

Value	Name	Description
0	NORMAL	Normal mode of operation.
1	OFFSET_ERROR	Offset Error mode to measure the offset error. See Table 61-7.
2	GAIN_ERROR_HIGH	Gain Error mode to measure the gain error. See Table 61-7.
3	GAIN_ERROR_LOW	Gain Error mode to measure the gain error. See Table 61-7.

See Section 61.6.14 “Automatic Error Correction” for details on ADC running mode.

61.7.17 ADC Compare Window Register

Name: ADC_CWR

Address: 0xFC030044

Access: Read/Write

31	30	29	28	27	26	25	24
		HIGHTHRES					
23	22	21	20	19	18	17	16
HIGHTHRES							
15	14	13	12	11	10	9	8
		LOWTHRES					
7	6	5	4	3	2	1	0
LOWTHRES							

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

- **LOWTHRES: Low Threshold**

Low threshold associated to compare settings of ADC_EMR.

- **HIGHTHRES: High Threshold**

High threshold associated to compare settings of ADC_EMR.

61.7.18 ADC Channel Offset Register

Name: ADC_COR

Address: 0xFC03004C

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	DIFF11	DIFF10	DIFF9	DIFF8
23	22	21	20	19	18	17	16
DIFF7	DIFF6	DIFF5	DIFF4	DIFF3	DIFF2	DIFF1	DIFF0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

- **DIFFx: Differential Inputs for Channel x**

0: Corresponding channel is set in Single-ended mode.

1: Corresponding channel is set in Differential mode.

61.7.19 ADC Channel Data Register

Name: ADC_CDRx [x=0..11]

Address: 0xFC030050

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
		DATA					
7	6	5	4	3	2	1	0
DATA							

- **DATA: Converted Data**

The analog-to-digital conversion data is placed into this register at the end of a conversion and remains until a new conversion is completed. ADC_CDRx is only loaded if the corresponding analog channel is enabled.

61.7.20 ADC Analog Control Register

Name: ADC_ACR

Address: 0xFC030094

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	IBCTL	
7	6	5	4	3	2	1	0
–	–	–	–	–	–	PENDETSENS	

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

- **PENDETSENS: Pen Detection Sensitivity**

Modifies the pen detection input pull-up resistor value. See section “Electrical Characteristics” for further details.

- **IBCTL: ADC Bias Current Control**

Adapts performance versus power consumption. See section “Electrical Characteristics” for further details.

61.7.21 ADC Touchscreen Mode Register

Name: ADC_TSMR

Address: 0xFC0300B0

Access: Read/Write

31	30	29	28	27	26	25	24
PENDBC				–	–	–	PENDET
23	22	21	20	19	18	17	16
–	NOTSDMA	–	–	TSSCTIM			
15	14	13	12	11	10	9	8
–	–	–	–	TSFREQ			
7	6	5	4	3	2	1	0
–	–	TSAV		–	–	TSMODE	

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

- **TSMODE: Touchscreen Mode**

Value	Name	Description
0	NONE	No Touchscreen
1	4_WIRE_NO_PM	4-wire Touchscreen without pressure measurement
2	4_WIRE	4-wire Touchscreen with pressure measurement
3	5_WIRE	5-wire Touchscreen

When TSMOD equals 01 or 10 (i.e., 4-wire mode), channels 0, 1, 2 and 3 must not be used for classic ADC conversions. When TSMOD equals 11 (i.e., 5-wire mode), channels 0, 1, 2, 3, and 4 must not be used.

- **TSAV: Touchscreen Average**

Value	Name	Description
0	NO_FILTER	No Filtering. Only one ADC conversion per measure
1	AVG2CONV	Averages 2 ADC conversions
2	AVG4CONV	Averages 4 ADC conversions
3	AVG8CONV	Averages 8 ADC conversions

- **TSFREQ: Touchscreen Frequency**

Defines the touchscreen frequency compared to the trigger frequency.

TSFREQ must be greater or equal to TSAV.

The touchscreen frequency is:

$$\text{Touchscreen Frequency} = \text{Trigger Frequency} / 2^{\text{TSFREQ}}$$

- **TSSCTIM: Touchscreen Switches Closure Time**

Defines closure time of analog switches necessary to establish the measurement conditions.

The closure time is:

$$\text{Switch Closure Time} = (\text{TSSCTIM} \times 4) \text{ ADCCLK periods.}$$

- **PENDET: Pen Contact Detection Enable**

0: Pen contact detection disabled.

1: Pen contact detection enabled.

When PENDET = 1, XPOS, YPOS, Z1, Z2 values of ADC_XPOSR, ADC_YPOSR, ADC_PRESSR are automatically cleared when PENS = 0 in ADC_ISR.

- **NOTSDMA: No TouchScreen DMA**

0: XPOS, YPOS, Z1, Z2 are transmitted in ADC_LCDR.

1: XPOS, YPOS, Z1, Z2 are never transmitted in ADC_LCDR, therefore the buffer does not contains touchscreen values.

- **PENDBC: Pen Detect Debouncing Period**

Debouncing period = 2^{PENDBC} ADCCLK periods.

61.7.22 ADC Touchscreen X Position Register

Name: ADC_XPOSR

Address: 0xFC0300B4

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	XSCALE			
23	22	21	20	19	18	17	16
XSCALE							
15	14	13	12	11	10	9	8
–	–	–	–	XPOS			
7	6	5	4	3	2	1	0
XPOS							

- **XPOS: X Position**

The position measured is stored here. If XPOS = 0 or XPOS = XSIZE, the pen is on the border.

When pen detection is enabled (PENDET set to ‘1’ in ADC_TSMR), XPOS is tied to 0 while there is no detection of contact on the touchscreen (i.e., when PENS bit is cleared in ADC_ISR).

- **XSCALE: Scale of XPOS**

Indicates the max value that XPOS can reach. This value should be close to 2^{12}.

61.7.23 ADC Touchscreen Y Position Register

Name: ADC_YPOSR

Address: 0xFC0300B8

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	YSCALE			
23	22	21	20	19	18	17	16
YSCALE							
15	14	13	12	11	10	9	8
–	–	–	–	YPOS			
7	6	5	4	3	2	1	0
YPOS							

- **YPOS: Y Position**

The position measured is stored here. If YPOS = 0 or YPOS = YSIZE, the pen is on the border.

When pen detection is enabled (PENDET set to ‘1’ in ADC_TSMR), YPOS is tied to 0 while there is no detection of contact on the touchscreen (i.e., when PENS bit is cleared in ADC_ISR).

- **YSCALE: Scale of YPOS**

Indicates the max value that YPOS can reach. This value should be close to 2^{12}.

61.7.24 ADC Touchscreen Pressure Register

Name: ADC_PRESSR

Address: 0xFC0300BC

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	Z2			
23	22	21	20	19	18	17	16
Z2							
15	14	13	12	11	10	9	8
–	–	–	–	Z1			
7	6	5	4	3	2	1	0
Z1							

- **Z1: Data of Z1 Measurement**

Data Z1 necessary to calculate pen pressure.

When pen detection is enabled (PENDET set to ‘1’ in ADC_TSMR), Z1 is tied to 0 while there is no detection of contact on the touchscreen (i.e., when PENS bit is cleared in ADC_ISR).

- **Z2: Data of Z2 Measurement**

Data Z2 necessary to calculate pen pressure.

When pen detection is enabled (PENDET set to ‘1’ in ADC_TSMR), Z2 is tied to 0 while there is no detection of contact on the touchscreen (i.e., when PENS bit is cleared in ADC_ISR).

Note: These two values are unavailable if TSMODE is not set to 2 in ADC_TSMR.

61.7.25 ADC Trigger Register

Name: ADC_TRGR

Address: 0xFC0300C0

Access: Read/Write

31	30	29	28	27	26	25	24
TRGPER							
23	22	21	20	19	18	17	16
TRGPER							
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	TRGMOD		

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

- **TRGMOD: Trigger Mode**

Value	Name	Description
0	NO_TRIGGER	No trigger, only software trigger can start conversions
1	EXT_TRIG_RISE	External trigger rising edge
2	EXT_TRIG_FALL	External trigger falling edge
3	EXT_TRIG_ANY	External trigger any edge
4	PEN_TRIG	Pen Detect Trigger (shall be selected only if PENDET is set and TSAMOD = Touchscreen only mode)
5	PERIOD_TRIG	ADC internal periodic trigger (see field TRGPER)
6	CONTINUOUS	Continuous mode

- **TRGPER: Trigger Period**

Effective only if TRGMOD defines a periodic trigger.

Defines the periodic trigger period, with the following equation:

Trigger Period = (TRGPER + 1) / ADCCLK

The minimum time between two consecutive trigger events must be strictly greater than the duration time of the longest conversion sequence depending on the configuration of registers ADC_MR, ADC_CHSR, ADC_SEQRx, ADC_TSMR.

When TRGMOD is set to pen detect trigger (i.e., 100) and averaging is used (i.e., field TSAV ≠ 0 in ADC_TSMR) only one measure is performed. Thus, XRDY, YRDY, PRDY, DRDY will not rise on pen contact trigger. To achieve measurement, several triggers must be provided either by software or by setting the TRGMOD on continuous trigger (i.e., 110) until flags rise.

61.7.26 ADC Correction Values Register

Name: ADC_CVR

Address: 0xFC0300D4

Access: Read/Write

31	30	29	28	27	26	25	24
GAINCORR							
23	22	21	20	19	18	17	16
GAINCORR							
15	14	13	12	11	10	9	8
OFFSETCORR							
7	6	5	4	3	2	1	0
OFFSETCORR							

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

- **OFFSETCORR: Offset Correction**

Offset correction to apply on converted data. The offset is signed (2’s complement), only bits 0 to 11 are relevant (other bits are ignored and read as 0).

- **GAINCORR: Gain Correction**

Gain correction to apply on converted data. Only bits 0 to 13 are relevant (other bits are ignored and read as 0).

61.7.27 ADC Channel Error Correction Register

Name: ADC_CECR

Address: 0xFC0300D8

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	ECORR11	ECORR10	ECORR9	ECORR8
7	6	5	4	3	2	1	0
ECORR7	ECORR6	ECORR5	ECORR4	ECORR3	ECORR2	ECORR1	ECORR0

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

- **ECORRx: Error Correction Enable for channel x**

0: Automatic error correction is disabled for channel x.

1: Automatic error correction is enabled for channel x.

61.7.28 ADC Touchscreen Correction Values Register

Name: ADC_TSCVR

Address: 0xFC0300DC

Access: Read/Write

31	30	29	28	27	26	25	24
TSGAINCORR							
23	22	21	20	19	18	17	16
TSGAINCORR							
15	14	13	12	11	10	9	8
TSOFFSETCORR							
7	6	5	4	3	2	1	0
TSOFFSETCORR							

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

- **TSOFFSETCORR: Touchscreen Offset Correction**

Offset correction to apply on converted data for the touchscreen channels. The offset is signed (2’s complement), only bits 0 to 11 are relevant (other bits are ignored and read as 0).

- **TSGAINCORR: Touchscreen Gain Correction**

Gain correction to apply on converted data for the touchscreen channels. Only bits 0 to 13 are relevant (other bits are ignored and read as 0).

61.7.29 ADC Write Protection Mode Register

Name: ADC_WPMR

Address: 0xFC0300E4

Access: Read/Write

31	30	29	28	27	26	25	24
WPKEY							
23	22	21	20	19	18	17	16
WPKEY							
15	14	13	12	11	10	9	8
WPKEY							
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	WPEN

- **WPEN: Write Protection Enable**

0: Disables the write protection if WPKEY value corresponds to 0x414443 (“ADC” in ASCII).

1: Enables the write protection if WPKEY value corresponds to 0x414443 (“ADC” in ASCII).

See Section 61.6.19 “Register Write Protection” for the list of write-protected registers.

- **WPKEY: Write Protection Key**

Value	Name	Description
0x414443	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0

61.7.30 ADC Write Protection Status Register

Name: ADC_WPSR

Address: 0xFC0300E8

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
WPVSRC							
15	14	13	12	11	10	9	8
WPVSRC							
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	WPVS

- **WPVS: Write Protection Violation Status**

0: No write protection violation has occurred since the last read of ADC_WPSR.

1: A write protection violation has occurred since the last read of ADC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

- **WPVSRC: Write Protection Violation Source**

When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

62. Electrical Characteristics

62.1 Absolute Maximum Ratings

Table 62-1. Absolute Maximum Ratings*

Storage Temperature.....-60°C to +150°C
Voltage on Input Pins with Respect to Ground..... -0.3V to +4.0V
Maximum Operating Voltage VDDCORE, VDDPLLA, VDDUTMIC and VDDHSIC.....1.5V
VDDIODDR.....2.0V
VDDBU.....4.0V
VDDIOPx, VDDUTMII, VDDISC, VDDSDMMC, VDDOSC, VDDANA, VDDAUDIOPLL.....4.0V
VDDFUSE.....3.0V
Total DC Output Current on all I/O lines.....350 mA

*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

62.2 DC Characteristics

The following characteristics are applicable to the operating temperature range T_A = -40°C to +105°C, unless otherwise specified.

Table 62-2. Recommended Thermal Operating Conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
T_A	Operating Temperature	–	-40	–	+105	°C
T_J	Junction Temperature	–	-40	–	+125	°C
R_{thJA}	Junction-to-ambient thermal resistance	LFBGA289	–	34.8	–	°C/W
		TFBGA256	–	27.4	–	
		TFBGA196	–	44.6	–	
R_{thJC}	Junction-to-case thermal resistance	LFBGA289	–	10.9	–	°C/W
		TFBGA256	–	TBD	–	
		TFBGA196	–	11.2	–	
P_D	Power Dissipation	At T_A = 85°C, LFBGA289	–	–	1149	mW
		At T_A = 105°C, LFBGA289	–	–	575	mW
P_D	Power Dissipation	At T_A = 85°C, TFBGA256	–	–	1460	mW
		At T_A = 105°C, TFBGA256	–	–	730	mW
P_D	Power Dissipation	At T_A = 85°C, TFBGA196	–	–	897	mW
		At T_A = 105°C, TFBGA196	–	–	448	mW

Table 62-3. DC Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDCORE}	DC Supply Core	–	1.1	1.2	1.32	V
	Allowable Voltage Ripple	rms value 10 kHz to 20 MHz	–	–	15	mV
	Slope	–	1.3	–	–	V/ms
V_{DDBU}	DC Supply I/Os, Backup	Must be established first	1.65		3.6	V
	Allowable Voltage Ripple	rms value 10 kHz to 10 MHz	–	–	30	mV
	Slope	–	2.4	–	–	V/ms
V_{DDANA}	DC Supply I/Os, Backup	The ADC is not functional below 2.0V	1.65		3.6	V
	Allowable Voltage Ripple	rms value 10 kHz to 20 MHz	–	–	20	mV
	Slope	–	2.4	–	–	V/ms
V_{DDIOP0}	DC Supply LCD I/Os		1.65	–	3.6	V
	Allowable Voltage Ripple	rms value 10 kHz to 20 MHz	–	–	20	mV
V_{DDIOP1}	DC Supply Peripheral I/Os	Peripheral I/O Lines	1.65	–	3.6	V
	Allowable Voltage Ripple	rms value 10 kHz to 20 MHz	–	–	20	mV
V_{DDIOP2}	DC Supply Peripheral I/Os	Peripheral I/O Lines	1.65	–	3.6	V
	Allowable Voltage Ripple	rms value 10 kHz to 20 MHz	–	–	20	mV
V_{DDPLLA}	PLL A and Main Oscillator Supply	–	1.1	1.2	1.32	V
	Allowable Voltage Ripple	rms value 10 kHz to 10 MHz	–	–	20	mV
		rms value > 10 MHz	–	–	20	
$V_{DDUTMIC}$	DC Supply UDPHS and UHPHS UTMI+ Core	–	1.1	1.2	1.32	V
	Allowable Voltage Ripple	rms value 10 kHz to 10 MHz	–	–	20	mV
$V_{DDUTMII}$	DC Supply UDPHS and UHPHS UTMI+ Interface	–	3.0	3.3	3.6	V
	Allowable Voltage Ripple	rms value 10 kHz to 10 MHz	–	–	20	mV
V_{DDHSIC}	DC Supply HSIC Phy	–	1.1	1.2	1.3	V
	Allowable Voltage Ripple	rms value 10 kHz to 10 MHz	–	–	20	mV
$V_{DDAUDIOPLL}$	DC Supply AUDIO PLL	–	3.0	3.3	3.6	V
	Allowable Voltage Ripple	rms value 10 kHz to 10 MHz	–	–	20	mV
V_{DDFUSE}	DC Supply Fuse Box	For fuse programming only	2.25	2.5	2.75	V
	Allowable Voltage Ripple	rms value 10 kHz to 10 MHz	–	–	20	mV
$V_{DDSDMMC}$	DC Supply Peripheral I/Os	SDMMC I/Os Lines	1.65	–	3.6	V
	Allowable Voltage Ripple	rms value 10 kHz to 10 MHz	–	–	20	mV

Table 63-3. DC Characteristics (Continued)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDISC}	DC Supply Peripheral I/Os	ISC I/Os Lines	1.65	–	3.6	V
	Allowable Voltage Ripple	rms value 10 kHz to 10 MHz	–	–	20	mV
V_{DDOSC}	DC Supply Oscillator	–	1.65	–	3.6	V
	Allowable Voltage Ripple	rms value 10 kHz to 10 MHz	–	–	15	mV
$V_{DDIODDR}$	DC Supply SDRAM I/Os	- LPDDR1-DDR2 Interface I/O lines - LPDDR2-LPDDR3 Interface I/O lines - DDR3L Interface I/O lines - DDR3 Interface I/O lines	1.7 1.14 1.283 1.425	1.8 1.2 1.35 1.5	1.95 1.30 1.45 1.575	V
V_{IL}	Low-level Input Voltage(2)	VDDIO in 3.3V range	-0.3	–	0.8	V
		VDDIO in 1.8V range	-0.3	–	0.3 x V_{DDIO}	
V_{IH}	High-level Input Voltage(2)	VDDIO in 3.3V range	2	–	V_{DDIO} +0.3	V
		VDDIO in 1.8V range	0.7 x V_{DDIO}	–	V_{DDIO} +0.3	
V_{OL}	Low-level Output Voltage	I_O MAX	–	–	0.41	V
V_{OH}	High-level Output Voltage	I_O MAX	V_{DDIO} - 0.4	–	–	V
V_{hys}	Schmitt Trigger Hysteresis	All PIO lines, VDDIOx in 3.3V range	0.34	–	–	V
		All PIO lines, VDDIOx in 1.8V range	0.2	–	–	
I_{OL}	I_{OIL}(or I_{SINK}) (VOL = 0.4V)	All GPIO_x, 1.8V: Low	-1	–	–	mA
		All GPIO_x, 1.8V: Medium	-10	–	–	
		All GPIO_x, 1.8V: High	-18	–	–	
I_{OH}	I_{OH} (or I_{SOURCE}) (VOH = VDDIO - 0.4V)	All GPIO_x, 1.8V: Low	–	–	1	mA
		All GPIO_x, 1.8V: Medium	–	–	10	
		All GPIO_x, 1.8V: High	–	–	18	
I_{OL}	I_{OIL}(or I_{SINK}) (VOL = 0.4V)	All GPIO_x, 3.3V: Low	-2	–	–	mA
		All GPIO_x, 3.3V: Medium	-20	–	–	
		All GPIO_x, 3.3V: High	-32	–	–	
I_{OH}	I_{OH} (or I_{SOURCE}) (VOH = VDDIO - 0.4V)	All GPIO_x, 3.3V: Low	–	–	2	mA
		All GPIO_x, 3.3V: Medium	–	–	20	
		All GPIO_x, 3.3V: High	–	–	32	
I_{IL}	Low-level Input Current	LCDPCK, ISI_MCK, GPIO, QSPI_SCK	0	–	0.1	µA
			10	–	70	
I_{IH}	High-level Input Current	LCDPCK, ISI_MCK, GPIO, QSPI_SCK	0	–	0.1	µA
			10	–	60	
I_{IL}	Low-level Input Current	GPIO_AD	0	–	0.1	µA
			5	–	10	
I_{IH}	High-level Input Current	GPIO_AD	0	–	0.1	µA
			5	–	10	

Table 63-3. DC Characteristics (Continued)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
R_{PULLUP}	Pull-up Resistor	GPIO_CLK, GPIO_IO, GPIO: 1.8V	80	143	310	kΩ
		GPIO_CLK, GPIO_IO, GPIO: 3.3V	40	66	130	
$R_{PULLDOWN}$	Pull-down Resistor	GPIO_CLK, GPIO_IO, GPIO: 1.8V	80	161	430	kΩ
		GPIO_CLK, GPIO_IO, GPIO: 3.3V	40	77	160	
R_{PULLUP}	Pull-up Resistor	GPIO_AD: 1.8V	280	380	480	kΩ
		GPIO_AD: 3.3V	280	380	480	
$R_{PULLDOWN}$	Pull-down Resistor	GPIO_AD: 1.8V	280	380	480	kΩ
		GPIO_AD: 3.3V	280	380	480	
R_{SERIAL}	Serial Resistor	GPIO	–	30	–	Ω
		GPIO_IO	–	13	–	Ω
		GPIO_CLK, GPIO_AD	–	0	–	Ω

Notes: 1. V_{DDIO} voltage must be equal to V_{DDIN} voltage.
2. Current injection may lead to performance degradation or functional failures.

Table 62-4. I/O Switching Frequency

GPIO Type	Drive	VDD			Unit
	C_{Load} = 30pF	1.8V	2.5V	3.3V	
GPIO_IO	Low	8	12	15	MHz
	Medium	60	80	90	
	High	80	110	110	
GPIO_CLK	Low	10	15	18	
	Medium	90	100	120	
	High	–	–	–	
GPIO_AD	Low	10	15	18	
	Medium	90	100	120	
	High	–	–	–	
GPIO	Low	7	10	12	
	Medium	40	50	60	
	High	50	60	70	

Table 62-5. QSPI I/O Switching Frequency

GPIO Type	Description	Name	Conditions	Min	Max	Unit
GPIO_QSPI	Maximum output frequency	f_{max}	Load = 30 pF	–	133	MHz
	Output duty cycle	–	Load = 30 pF	45	55	%

62.3 Power Consumption

This section provides information about the current consumption on different power supply rails of the device. It gives current consumption in:

- Active mode when running a CoreMark and in predefined use cases
- Low-power modes: Backup mode, Idle mode and Ultra Low-power mode
- By peripheral with representative activity

62.4 Active Mode

Active mode is the normal running mode with the ARM core clock running off a PLL. The power management controller is used to adapt the frequency and to disable the peripheral clocks.

62.4.1 Active Mode Power Consumption Versus Modes

The power consumption values are measured under the following operating conditions:

- Parts are from typical process
- V_{DDIOPx} = 3.3V
- $V_{DDSDMMC0}$ and $V_{DDSDMMC1}$ = 1.8V to 3.3V (high frequency)
- V_{DDCORE} = 1.2V +/-2%
- V_{DDBU} = 1.6V to 3.6V
- T_A = as specified in Table 62-6 and Table 62-7
- There is no consumption on the device's I/Os.
- All peripheral clocks are disabled.

Figure 62-1. Measurement Schematics

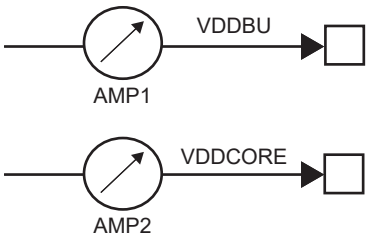


Table 62-6. Typical Peripheral Power Consumption by Peripheral in Active Mode

Peripheral	Conditions (T_A = 25°C)	Consumption on VDDCORE	Conditions	Consumption (typ)	Unit
GMAC	Peripheral Clock Enabled	–	See Note 1.	12 *MCK + 1840*DR (Data Rate in Mbits/s)	µA
XDMAC0		–	See Note 2.	17.6 * MCK + 4.92 * DR (MCK in MHz, DataRate in Mbytes/s)	
XDMAC1		–			
ICM		3.52	–	–	µA/MHz
AES		8.79			
AESB		7.49			
TDES		0.85			
SHA		3.88			
MPDDRC		45.21	See Note 3.	67 * MCK + 3.11 * DR + 1609 (MCK in MHz, DataRate in Mbytes/s)	µA
HSMC		20.12	–	–	µA/MHz
PIOA		11.39			
FLEXCOM0		-			
FLEX0_USART		5.21			
FLEX0_SPI		7.39			
FLEX0_TWI		3.88			
FLEXCOM1		See FLEXCOM0			
FLEXCOM2		See FLEXCOM0			
FLEXCOM3		See FLEXCOM0			
FLEXCOM4		See FLEXCOM0			
UART0		1.09			
UART1		0.97			
UART2		1.21			
UART3		0.85			
UART4		0.85			
TWIHS0		3.27			
TWIHS1		3.39			
SDMMC0		8.61			
SDMMC1		8.61			
SPI0		4.61			
SPI1		4.48			
TC0		3.03			
TC1		4			
PWM		7.03			
ADC		2.3			

Table 63-6. Typical Peripheral Power Consumption by Peripheral in Active Mode (Continued)

Peripheral	Conditions (T_A = 25°C)	Consumption on VDDCORE	Conditions	Consumption (typ)	Unit
UHPHS	Peripheral Clock Enabled	–	See Note 4.	12 *MCK + 4900*DR (MCK in MHz, Data Rate in Mbytes/s)	µA
UDPHS			See Note 5.	10 *MCK + 2060*DR (MCK in MHz, Data Rate in Mbytes/s)	
SSC0		1.58	–	–	µA/MHz
SSC1		1.58			
LCDC		–	See Note 6.	9.8 *MCK + 32 *Pix_CLK + 15.7*DR_Baselayer +30.1*DR_Overlayer (MCK & Pixelclock in MHz, DataRate in Mbytes/s);	µA
ISC			See Note 7.	10.9*MCK + 22.4*ISPCK + 12.38*DR_sensor (MCK&ISPCK in MHz, DataRate in Mbytes/s)	
TRNG		760	–	–	
PDMIC		8.24	–	–	µA/MHz
QSPI0		1.94			
QSPI1		1.94			
I2SC0		0.61			
I2SC1		0.61			
CAN0		9.7			
CAN1		7.27			

Notes:

1. In Linux OS, use the ‘iperf’ command to perform bidirectional data transfers. Measure GMAC consumption at different transfer speeds.
2. XDMAC is initialized and one channel performs a memory-to-memory transfer. During test, the data rate is adjusted by changing the DMA setting and the burst size.
3. DDR3 devices are initialized (fully functional). XDMAC performs a memory-to-memory transfer inside the DDR area. Total consumption of MPDDRC and XDMAC is measured. MPDDRC consumption is calculated by discounting XDMAC consumption.
4. In Linux OS, measure UHPHS consumption at different transfer speeds.
5. In Linux OS, build a mass storage using UDPHS. Measure UDPHS consumption at different transfer
6. The LCD timing engine and each display layer are switched on in sequence. The static image (using random data) is displayed under various resolutions. The 24-bpp RGB888 color space is set for all layers. Auxiliary functions such as rotation, scaling, color space conversion, color look-up table, and chroma upsampling are disabled.
7. ISC performs image sensor preview.

In order to maximize performances, each Peripheral Clock has been timed to H32MX clock frequency. The peripheral frequency can be reduced with the help of a divider in PMC_PCR.

Table 62-7. Power Consumption in Active Mode: AMP2

Conditions T_A = 25°C		Consumption			
		Dhrystone (mA)	DMIPS	CoreMark (mA)	CoreMark
PLL clock is 1000 MHz, ARM Core clock is 500 MHz, MCK is 166 MHz. – Caches L1 and L2 enabled – Code running off of internal SRAM – Code speed optimization – Run Dhrystone / CoreMark benchmark – Peripheral clock disabled	MRL-B	114.4	1.277	108.8	2.65
	MRL-A	237.2	1.277	233.7	2.65

62.5 Low-power Modes

The various low-power modes enable balancing device power consumption and wakeup time. The modes are described below, in the order of lowest to highest power consumption.

62.5.1 Backup Mode

The Backup mode allows to achieve the lowest power consumption in the system with limited functionality.

In this mode, only the backup area is powered, maintaining the RTC, the backup registers, the backup SRAM and the security module running. This mode is entered by shutting down all the power rails except the VDDBU (refer to Section 23. “Shutdown Controller (SHDWC)”). To exit Backup mode, the SHDN pin (connected to the enable of the external Power Management IC) must be driven high by an internal event (RTC) or by one of the external events listed below:

- WKUP0 to WKUP9 pins (level transition, configurable debouncing)
- Character received on a serial com receiver (RXLP)
- Analog comparison

The Backup Mode functionality has been extended with the possibility to keep the DDR memory in self-refresh state.

62.5.2 Backup Mode with DDR in Self-refresh

The Backup mode with DDR in self-refresh is used to keep the DDR contents when the system is powered off. This mode is achieved by maintaining the backup area and the VDDIODDR powered. The sequence below must be performed to enter the Backup Self-refresh mode:

- Software saves all the context information to resume (application-dependent).
- Put the DDR in Self-refresh mode and wait until the self-refresh status is OK (refer to Section 33. “Multiport DDR-SDRAM Controller (MPDDRC)”).
- Set SFRBU_DDRBUMCR.BUMEN (Section 19.3.3 “DDR BU Mode Control Register”)
- Enter the Backup mode as described above.

The method to exit this mode is the same as described for the Backup mode. Once the system is restarted, the software checks the state of the SFRBU_DDRBUMCR.BUMEN bit and reinitializes the DDR controller. The DDR memory exits Self-refresh mode when a memory access in the DDR memory space is performed.

62.5.3 Ultra Low-power (ULP) Mode

The purpose of the Ultra Low-power mode is to achieve the lowest power consumption with the system in Retention mode and able to resume on wake up events (any interrupt or hardware event). This mode is a combination of the Wait for Interrupt mode of the ARM core and the system clocks frequency reduced or shut-off.

To obtain the best results, care must be taken that the I/Os (pull-up/pull-down, etc.), USB transceivers, etc., are set to the appropriate state.

The Ultra Low-power mode features two submodes:

- ULP0 mode
- ULP1 mode

62.5.3.1 ULP0 Mode

The ULP0 mode maintains a very low frequency clock to wake up on any interrupt.

The selection of the clock depends on the current consumption target versus wake up time. The higher the frequency, the higher the power consumption.

The sequence to enter ULP0 mode is detailed below. The code used to enter this mode must be executed out of the internal SRAM.

1. Set the DDR to Self-Refresh mode.
2. Set the interrupts to wake up the system.
3. Disable all peripheral clocks.
4. Set the I/Os to an appropriate state and disable the USB transceivers (refer to section SFR).
5. Switch the system clock to Slow Clock.
6. Disable the PLLs, the main oscillator and the 12 MHz RC oscillator.
7. Enter the Wait for Interrupt mode and disable the PCK clock in the PMC_SCDR.

The wake up from ULP0 mode is triggered by any enabled interrupt. When resuming, the software reconfigures the system (oscillator, PLL, etc.) in the same state as before WFI.

62.5.3.2 ULP1 Mode

Unlike the ULP0 mode, all the clocks are off in the ULP1 mode, but the number of wake up sources is limited to the list below:

- WKUP0 pin (level transition, configurable debouncing)
- WKUP1 Secumod wakeup signal
- WKUP2 pin to WKUP9 pin (shared with PIOBU0 to PIOBU7)
- RTC alarm
- USB Resume from Suspend mode
- SDMMC card detect
- RXLP event
- ACC event
- Any SleepWalking event coming from TWI, FLEXCOMx, SPI, ADC

The sequence to enter the ULP1 mode is detailed below. The code used to enter this mode must be executed out of the internal SRAM.

1. Set the DDR to Self-Refresh mode.
2. Set the events to enable a system wakeup.
3. Disable all peripheral clocks.
4. Set the I/Os to an appropriate state and disable the USB transceivers (refer to the section “Special Function Register (SFR)”).

5. Switch the system clock to the 12 MHz RC oscillator.
6. Disable the PLLs and the main oscillator.
7. Enter the ULP1 mode by either:
    1. setting the WAITMODE bit in CKGR_MOR, or
    2. setting the LPM bit in PMC_FSMR and executing the processor WaitForEvent (WFE) instruction.
8. After setting the WAITMODE bit or using the WFE instruction, wait for the PMC_SR.MCKRDY bit to be set.

62.5.4 Idle Mode

The purpose of Idle mode is to optimize power consumption of the device versus response time. In this mode, only the core clock is stopped. The peripheral clocks, including the DDR controller clock, can be enabled. The current consumption in this mode is application-dependent and can be reduced by enabling Dynamic Clock Gating (L2CC_POWCR.DCKGATEN = 1).

This mode is entered via the Wait for Interrupt (WFI) instruction and PCK disabling.

The processor can be awakened from an interrupt. The system resumes where it was before entering WFI mode.

62.5.5 Low-power Mode Summary Table

The modes detailed above are the main low-power modes. Each part can be set to on or off separately and wakeup sources can be configured individually. Table 62-8 shows a summary of the low-power mode configurations.

Table 62-8. Low-power Mode Configuration Summary

	Low-power Mode				
	Backup		Ultra-low-power		
Submode:	–	Self-refresh	ULP0	ULP1	Idle
64 kHz RC Oscillator, 32 kHz Oscillator, RTC, Backup Memory and Registers, POR	ON				
12 MHz RC Oscillator	OFF				ON
VDDCORE Regulator	OFF		ON		
Core	OFF (Not powered)		Powered (Not clocked)		
Memory, Peripherals	OFF (Not powered)		Powered (512 Hz)	Powered (Not clocked)	Powered (Clocked)
Mode Entry	Shutdown Controller, FLEXCOM SleepWalking	DDR in Self-refresh, Shutdown Controller	DDR in Self-refresh Frequency reduced in PMC, WFI	DDR in Self-refresh, CKGR_MOR.WAITMODE = 1	DDR in Self-refresh, WFI
Potential Wakeup Sources	WKUP0 pin, any PIOBU configured as WKUP pin, RTC alarm, any level above comparator source or character received	Backup mode sources	Any interrupt	Wakeup pins, WOL	Any interrupt
Core at Wakeup	Reset		Clocked back at 512 Hz	Clocked back at 12 MHz	Clocked back at full speed
PIO State While in Low-power Mode	Reset		Previous state saved		
PIO State at Wakeup	Inputs with pull-ups	Unchanged			
Consumption[2]	I_{VDDBU}= 4.5 µA typ[3] at 25°C/3.0V	I_{VDDBU}= 4.5 µA typ[3] at 25°C/3.0V $I_{VDDIODDR}$ = 40 µA	0.21 mA at 25°C/1.1V 0.27 mA at 25°C/1.2V	0.17 mA at 25°C/1.1V 0.27 mA at 25°C/1.2V	28 mA at 25°C/1.2V[4]
Wakeup Time[1]	Startup time	Startup time	300 ms	15 µs	800 ns at 498 MHz

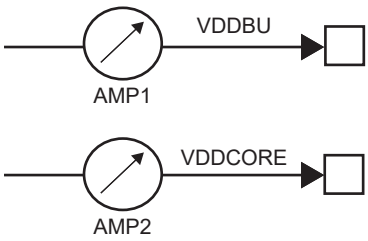
Notes:
1. When considering wakeup time, the time required to start the PLL is not taken into account. Once started, the device works with the main oscillator. The user has to add the PLL startup time if it is needed in the system. The wakeup time is defined as the time taken for wakeup until the first instruction is fetched.
2. The external loads on PIOs are not taken into account in the calculation.
3. Total current consumption.
4. Dynamic Clock Gating enabled (L2CC_POWCR.DCKGATEN = 1)

62.5.6 Low-power Consumption Versus Modes

The low-power consumption values are measured under the following operating conditions:

- Parts are from typical process
- V_{DDIOPx} = 3.3V
- $V_{DDSDMMC0}$ and $V_{DDSDMMC1}$ = 1.8V to 3.3V (high frequency)
- V_{DDCORE} = 1.2V +/-2%
- V_{DDBU} = 1.6V to 3.6V
- T_A = as specified in Table 62-9, Table 62-10, Table 62-11
- There is no consumption on the device's I/Os.
- All peripheral clocks are disabled.

Figure 62-2. Measurement Schematics



In order to maximize performances, each Peripheral Clock has been timed to H32MX clock frequency. The peripheral frequency can be reduced with the help of a divider in PMC_PCR.

Table 62-9. Typical Power Consumption in Idle Mode: AMP2

Conditions	Consumption				Unit
	T_A 25°C	T_A 70°C	T_A 85°C	T_A 105°C	
PLL clock is 1000 MHz, ARM Core clock is 500 MHz, MCK is 166 MHz. – Core clock is stopped – Peripheral clocks, including the DDR Controller clock, can be enabled – Mode is entered via Wait for Interrupt (WFI) instruction and PCK disabling – Measure IDDCORE + IDDBU – Peripheral clock disabled	28.16	29.63	30.82	33.44	mA

Table 62-10. VDDCORE Power Consumption in Ultra Low-power Mode: AMP2

Mode	Conditions	Consumption (mA)				Wakeup Time µs
		T_A 25°C	T_A 70°C	T_A 85°C	T_A 105°C	
ULP1 Fast Wakeup	ARM Core clock is disabled. MCK is 0.	0.27	1.44	2.38	4.63	15
ULP0 12 MHz	ARM Core clock is disabled. MCK is 12 MHz.	3.16	4.24	5.26	7.67	13
ULP0 750 kHz	ARM Core clock is disabled. MCK is 750 kHz.	1.55	2.62	3.68	6.06	205
ULP0 187 kHz	ARM Core clock is disabled. MCK is 187.5 kHz.	1.47	2.54	3.58	5.97	820
ULP0 32 kHz	ARM Core clock is disabled. MCK is 32 kHz.	0.28	1.52	2.59	5.03	4690
ULP0 512 Hz	ARM Core clock is disabled. MCK is 512 Hz.	0.275	1.51	2.58	5.025	300000

Table 62-11. Typical Power Consumption for Backup Mode

V_{DDBU} (V)	Conditions	Consumption (µA)				Unit
		T_A = 25°C	T_A = 70°C	T_A = 85°C	T_A = 105°C	
1.6	V_{DDBU} Only	4.2	12.1	19.3	36.8	µA
1.7		4.2	12.1	19.3	36.85	
1.8		4.25	12.1	19.35	36.85	
1.9		4.25	12.1	19.35	36.9	
2		4.3	12.1	19.4	36.95	
2.1		4.3	12.15	19.4	36.95	
2.2		4.3	12.2	19.45	37	
2.3		4.35	12.2	19.45	37	
2.4		4.35	12.2	19.5	37	
2.5		4.38	12.25	19.5	37.05	
2.6		4.4	12.25	19.55	37.1	
2.7		4.4	12.3	19.55	37.1	
2.8		4.4	12.3	19.6	37.15	
2.9		4.45	12.35	19.6	37.2	
3		4.45	12.35	19.65	37.3	
3.1		4.48	12.45	19.8	37.7	
3.2		4.55	12.8	20.25	38.3	
3.3		4.9	13.4	20.9	38.9	
3.4		5.5	14.1	21.6	39.7	
3.5		6.2	14.9	22.4	40.5	
3.6		7	15.7	23.3	41.4	

62.6 Clock Characteristics

62.6.1 Processor Clock Characteristics

Table 62-12. Processor Clock Waveform Parameters

Symbol	Parameter	Conditions	Min	Max	Unit
$1/(t_{CPPCK})$	Processor Clock Frequency	VDDCORE[1.1V, 1.32V] T = 85°C	250(1)	400	MHz
		VDDCORE[1.2V, 1.32V] T = 85°C	250(1)	500	

Note: 1. Limitation for DDR2 (125 MHz) usage only. There are no limitations to DDR3, DDR3L, LPDDR1, LPDDR2 and LPDDR3.

62.6.2 Master Clock Characteristics

The master clock is the maximum clock at which the system is able to run. It is given by the smallest value of the internal bus clock and EBI clock.

Table 62-13. Master Clock Waveform Parameters

Symbol	Parameter	Conditions	Min	Max	Unit
$1/(t_{CPMCK})$	Master Clock Frequency	VDDCORE[1.1V, 1.32V] T_A = 85°C	125(1)	133	MHz
		VDDCORE[1.2V, 1.32V], in DDR2 or LPDDR1 mode, VDDIODDR[1.8V, 1.95V], T_A = 85°C in LPDDR2 or LPDDR3 mode, VDDIODDR[1.2V, 1.30V], T_A = 85°C in DDR3 mode, VDDIODDR[1.5V, 1.575V], T_A = 85°C in DDR3L mode, VDDIODDR[1.35V, 1.45V], T_A = 85°C Security disabled	125(1)	166(2)	

Notes: 1. Limitation for DDR2 usage only. There are no limitations to DDR3, DDR3L, LPDDR1, LPDDR2 and LPDDR3.
2. The JEDEC standard specifies a maximum clock frequency of 125 MHz for DDR3 and DDR3L in DLL Off mode. However, check with memory suppliers for higher frequencies.

62.7 Oscillator Characteristics

62.7.1 Main Oscillator Characteristics

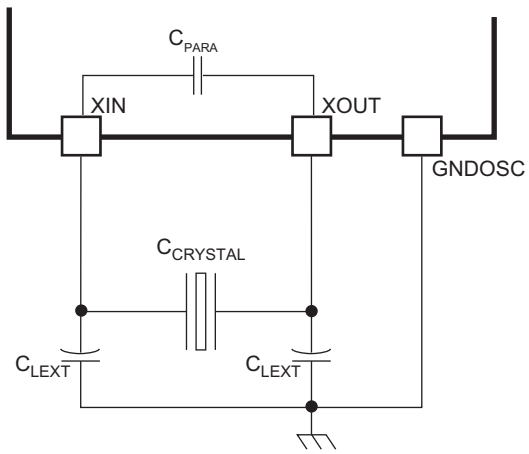
Table 62-14. 8 to 24 MHz Crystal Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OSC}	Operating Frequency	FREQ[2] = 00, 11 FREQ = 01 FREQ = 10	8 12 16	–	12 16 24	MHz
–	Duty Cycle	–	40	50	60	%
t_{START}	Startup Time	FREQ[2] = 00, 11 FREQ = 01 FREQ = 10	–	–	18.5 10.5 6	ms
I_{DDON}	Current Consumption (on VDDIO)	@ 12 MHz @ 24 MHz	–	1.2 1.7	3.5 4	mA
I_{DD_STDBY}	Standby Current	–	–	0.02	0.1	µA
C_{PARA}	Internal Parasitic Capacitance[1]	From XIN to XOUT	1.4	1.6	1.8	pF
P_{ON}	Drive level	FREQ = 00 FREQ = 01, 11 FREQ = 10	–	–	150 300 400	µW

Notes:
1. The external capacitors value can be determined by using the following formula:
   $C_{LEXT} = (2 \times C_{CRYSTAL}) - C_{BOARD} - (C_{PARA} \times 2)$
   where:
   C_{LEXT}: external capacitor value which must be soldered from XIN to GND and XOUT to GND
   $C_{CRYSTAL}$: crystal targeted load
   C_{BOARD}: external board parasitic capacitance (from XIN to GND or XOUT to GND)
   C_{PARA}: internal parasitic capacitance

2. The SFR_UTMICKTRIM.FREQ field defines the input frequency for the UTMI and the main oscillator. It is important to select the correct FREQ value because this has a direct influence on USB frequency.

Figure 62-3. Main Oscillator Schematics



62.7.1.1 Recommended Crystal Characteristics

The following characteristics are applicable to the operating temperature range T_A = -40°C to 85°C and to the worst case of power supply, unless otherwise specified.

Table 62-15. Recommended Crystal Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
ESR	Equivalent Series Resistance	FREQ = 00, 11	–	–	100	Ω
		FREQ = 10, 01	–	–	80	
CM	Motional Capacitance	FREQ = 00	5	–	9	fF
		FREQ = 01, 10, 11	1.3	–	3.2	
CS	Shunt Capacitance	FREQ = 00, 01, 10	–	–	3	pF
		FREQ = 11	–	–	1.3	
$C_{CRYSTAL}$	Allowed crystal capacitive load	From crystal specification FREQ = 00, 01, 11 FREQ = 10	 12.5 8	–	 18 12.5	pF

62.7.1.2 XIN Clock Characteristics

Table 62-16. XIN Clock Electrical Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$1/(t_{CPXIN})$	XIN Clock Frequency	–	–	–	50	MHz
t_{CPXIN}	XIN Clock Period	–	20	–	–	ns
t_{CHXIN}	XIN Clock High Half-period	–	0.4 x t_{CPXIN}	–	0.6 x t_{CPXIN}	ns
t_{CLXIN}	XIN Clock Low Half-period	–	0.4 x t_{CPXIN}	–	0.6 x t_{CPXIN}	ns
CIN	XIN Input Capacitance	–	–	–	25	pF
RIN	XIN Pulldown Resistor	–	–	–	500	kΩ
VIN	XIN Voltage	–	V_{DDOSC}	–	V_{DDOSC}	V

Note: These characteristics apply only when the Main Oscillator is in Bypass mode (i.e., when MOSCEN = 0 and OSCBYPASS = 1 in CKGR_MOR). See “PMC Clock Generator Main Oscillator Register” in PMC section.

62.7.2 12 MHz RC Oscillator Characteristics

Table 62-17. 12 MHz RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OSC}	RC Oscillator Frequency	@ 25°C	11.63	–	12.12	MHz
t_{START}	Startup Time	–	–	–	15	µs
Duty	Duty Cycle	–	45	50	55	%
I_{DDON}	Current Consumption	After startup time	–	160	350	µA

62.7.3 32.768 kHz Crystal Oscillator Characteristics

Table 62-18. 32.768 kHz Crystal Oscillator Characteristics

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
f_{OSC}	Operating Frequency	Normal mode with crystal		–	32.768	–	kHz
t_{START}	Startup Time	C_m > 3fF		–	–	1200	ms
I_{DDON}	Current Consumption	ESR < 50k ohm	$C_{CRYSTAL32}$ = 12.5 pF	–	440	900	nA
			$C_{CRYSTAL32}$ = 6 pF		600	900	
		ESR < 100k ohm	$C_{CRYSTAL32}$ = 12.5 pF		800	1200	
			$C_{CRYSTAL32}$ = 6 pF		700	1200	
C_{PARA32}	Internal Parasitic Capacitance	Between XIN32 and XOUT32		1.4	1.6	1.8	pF

Figure 62-4. 32 kHz Oscillator Schematics

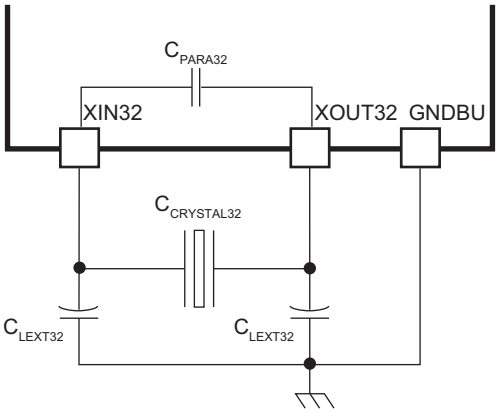


Table 62-19. Recommended 32.768 kHz Crystal Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
ESR	Equivalent Series Resistor	Crystal at 32.768 kHz	–	–	100	kΩ
–	Duty Cycle	–	40	50	60	%
C_m	Motional Capacitance	Crystal at 32.768 kHz	3	–	8	fF
C_{SHUNT}	Shunt Capacitance	Crystal at 32.768 kHz	0.6	–	2	pF
$C_{CRYSTAL32}$	Allowed Crystal Capacitance Load(1)	From crystal specification	6	–	12.5	pF
P_{ON}	Drive Level	–	–	–	0.2	µW

Note: 1. The external capacitors value can be determined by using the following formula:

$C_{LEXT32} = (2 \times C_{CRYSTAL32}) - C_{BOARD} - (C_{PARA32} \times 2)$

where:

C_{LEXT32}: external capacitor value which must be soldered from XIN32 to GND and XOUT32 to GND

$C_{CRYSTAL32}$: crystal targeted load.

C_{BOARD}: external board parasitic capacitance (from XIN to GND or XOUT to GND)

C_{PARA32}: internal parasitic capacitance

62.7.4 64 kHz RC Oscillator Characteristics

Table 62-20. 64 kHz RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OSC}	RC Oscillator Frequency	–	40		90	kHz
t_{START}	Startup Time	–	11	17	30	µs
I_{DDON}	Current Consumption	After startup time	–	93	140	nA

62.8 PLL Characteristics

Table 62-21. PLLA Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{IN}	Input Frequency	–	8	–	24	MHz
f_{OUT}	Output Frequency	–	600	–	1200	MHz
I_{PLL}	Current Consumption	Active mode	–	–	14.5	mA
		Standby mode	–	–	2	µA
t_{START}	Startup Time	–	–	–	60	µs

Table 62-22. UTMI PLL Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{IN}	Input Frequency	–	12	–	24	MHz
f_{OUT}	Output Frequency	–		480		MHz
$I_{VDDUTMII}$	Current Consumption	In Active mode, on VDDUTMII, @480 MHz	–	6.3	7.0	mA
t_{START}	Startup Time	–	–	–	60	µs

Table 62-23. Audio PLL Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{IN}	Input Frequency	–	12	–	24	MHz
$f_{AUDIOCORECLK}$	AUDIOCORECLK frequency range	–	620	–	700	MHz
$f_{AUDIOPINCLK}$	AUDIOPINCLK frequency range	–	8	12.288	48	MHz
$f_{AUDIOPLLCLK}$	AUDIOPLLCLK frequency range	–	–	–	150	MHz
I_{PLL}	Current Consumption	On VDDAUDIOPLL	6	–	20	mA
t_{START}	Startup Time	From OFF to stable AUDIOCORECLK frequency	–	–	100	µs
t_{SET}	Settling Time(1)	When changing FRACR or NR in PMC_AUDIO_PLL0 or PMC_AUDIO_PLL1	–	–	100	µs

Note: 1. Loop filter is set as recommended in fields BIAS_FILTER and DCO_FILTER of PMC_AUDIO_PLL0.

62.9 USB HS Characteristics

62.9.1 Electrical Characteristics

The device conforms to all voltage, power, and timing characteristics and specifications set forth in the USB 2.0 Specification. Refer to the USB 2.0 Specification for more information.

62.9.2 Dynamic Power Consumption

Table 62-24. USB Transceiver Dynamic Power Consumption

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
I_{BIAS}	Bias Generator Current Consumption	–	–	0.7	0.8	mA
$I_{VDDUTMII}$	HS Transceiver Current Consumption	HS transmission	–	47	60	mA
	HS Transceiver Current Consumption	HS reception	–	18	27	mA
	LS / FS Transceiver Current Consumption	FS transmission 0m cable(1)	–	4	6	mA
	LS / FS Transceiver Current Consumption	FS transmission 5m cable(1)	–	26	30	mA
	LS / FS Transceiver Current Consumption	FS reception(1)	–	3	4.5	mA
$I_{VDDUTMIC}$	Core	–	–	5.5	9	mA

Note: 1. Including 1 mA due to pull-up/pull-down current consumption.

62.10 ADC Characteristics

Electrical data are in accordance with an operating temperature range from -40°C to +85°C unless otherwise specified.

ADVREF is the positive reference of the ADC.

62.10.1 ADC Power Supply

62.10.1.1 Power Supply Characteristics

Table 62-25. Power Supply Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
I_{VDDIN}	Analog Current Consumption	Sleep mode(1) Fast Wakeup mode Normal mode, single sampling	-	2 0.4 2.2	4 0.6 3.0	µA mA mA
$I_{VDDCORE}$	Digital Current Consumption	Sleep mode(1) Normal mode	-	1 80	2 100	µA µA

Note: 1. In Sleep mode, the ADC core, the Sample and Hold and the internal reference operational amplifier are off.

62.10.2 External Reference Voltage

V_{ADVREF} is an external reference voltage applied on the pin ADVREF. The quality of the reference voltage V_{ADVREF} is critical to the performance of the ADC. A DC variation of the reference voltage V_{ADVREF} is converted to a gain error by the ADC. The noise generated by V_{ADVREF} is converted by the ADC to count noise.

Table 62-26. ADVREF Electrical Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{ADVREF}	Voltage Range	Full operational	2	–	VDDANA	V
	RMS Noise	Bandwidth 10 kHz to 1 MHz	–	–	100	µV
R_{ADVREF}	Input DC Impedance	ADC reference resistance bridge(1)	6	8	10	kΩ
I_{ADVREF}	Current	V_{ADVREF} = 3.3V	-	-	460	µA

Note: 1. When the ADC is off, the ADVREF impedance has a minimum of 1 MΩ.

62.10.3 ADC Timings

Table 62-27. ADC Timing Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{ADC_Clock}	Clock Frequency	–	0.2	–	20	MHz
f_S	Sampling Frequency(1)	–	–	–	1	MHz
t_{START}	ADC Startup Time	Sleep mode to Normal mode Fast Wakeup mode to Normal mode	–	–	4 2	µs

Note: 1. $t_{ADC_}$Clock = $1/f_{ADC\_Clock}$
ADC conversion time = 21 t_{ADC_Clock}
The Tracking time of the ADC has a minimal value of $t_{TRACKTIM}$ = 15 t_{ADC_Clock}.

62.10.4 ADC Transfer Function

The DATA code in ADC_CDR is up to 12-bit positive integer or two's complement (signed integer).

62.10.4.1 Differential Mode (12-bit mode)

A differential input voltage $V_{IN} = V_{INP} - V_{INN}$ can be applied between two selected differential pins, e.g. ADC0_AD0 and ADC0_AD1. The ideal code C_i is calculated by using the following formula and rounding the result to the nearest positive integer.

$$C_i = \frac{2047}{V_{ADVREF}} \times V_{IN}$$

For the other resolution defined by RES, the code C_i is extended to the corresponding resolution.

Table 62-28 is a computation example for the above formula, where V_{ADVREF} = 3V.

Table 62-28. Input Voltage Values in Differential Mode, Non-signed Output

Signed C_i	V_{IN}
-2048	-3
0	0
2047	3

62.10.4.2 Single-ended Mode (12-bit mode)

A single input voltage V_{IN} can be applied to selected pins, e.g., ADC0_AD0 or ADC0_AD1. The ideal code C_i is calculated using the following formula and rounding the result to the nearest positive integer.

The single-ended ideal code conversion formula is:

$$C_i = \frac{4095}{V_{ADVREF}} \times V_{IN}$$

For the other resolution defined by RES, the code C_i is extended to the corresponding resolution.

Table 62-29 is a computation example for the above formula, where V_{ADVREF} = 3V:

Table 62-29. Input Voltage Values in Single-ended Mode

Non-signed C_i	V_{IN}
0	0
2047	1.5
4095	3

62.10.4.3 Example of LSB Computation

The LSB is relative to the analog scale V_{ADVREF}.

The term LSB expresses the quantization step in volts, also used for one ADC code variation.

- Single-ended (SE) (ex: V_{ADVREF} = 3.0V)
  - Gain = 1, LSB = (3.0V / 4096) = 732 µV
- Differential (DIFF) (ex: V_{ADVREF} = 3.0V)
  - Gain = 0.5, LSB = (6.0V / 4096) = 1465 µV

The data include the ADC performances, as the PGA and ADC core cannot be separated. The temperature and voltage dependencies are given as separate parameters.

62.10.4.4 Gain and Offset Errors

For:

- a given gain error: E_G (%)
- a given ideal code (C_i)
- a given offset error: E_O (LSB of 12 bits)

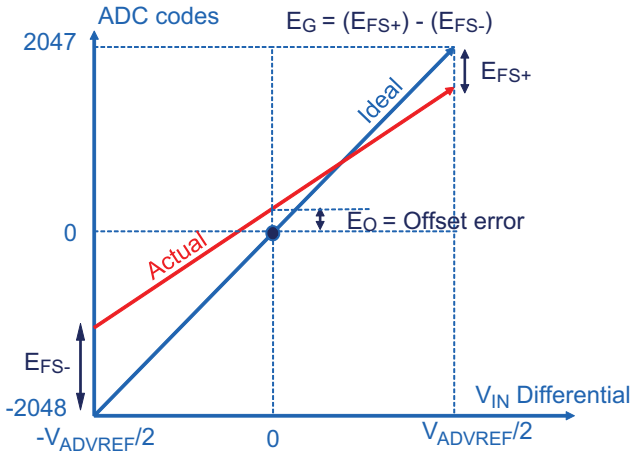
in 12-bit mode, the actual code (C_A) is calculated using the following formula

$$C_A = \left(1 + \frac{E_G}{100}\right) \times (C_i - 2047) + 2047 + E_O$$

Differential Mode

In Differential mode, the offset is defined when the differential input voltage is zero.

Figure 62-5. Gain and Offset Errors in Differential Mode



where:

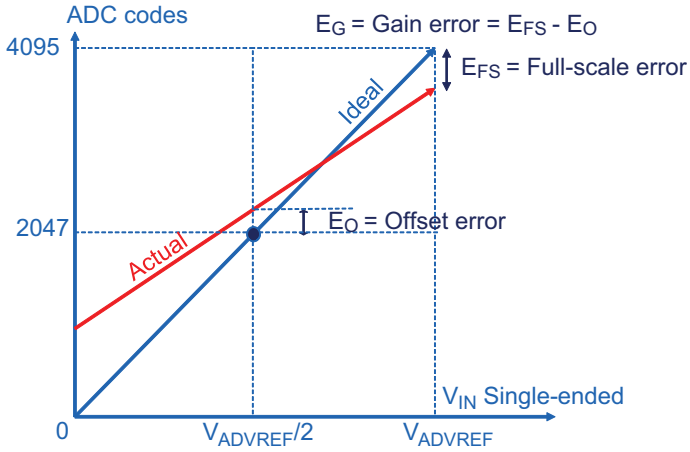
- Full-scale error $E_{FS} = (E_{FS+}) - (E_{FS-})$, unit is LSB code
- Offset error E_O is the offset error measured for V_{IN} = 0V
- Gain error $E_G = 100 \times E_{FS} / 4096$, unit in %

The error values in Table 62-31 include the sample-and-hold error as well as the PGA gain error.

Single-ended Mode

Figure 62-6 illustrates the ADC output code relative to an input voltage V_{IN} between 0V (Ground) and V_{ADVREF}. The ADC is configured in Single-ended mode by connecting internally the negative differential input to $V_{ADVREF} / 2$. As the ADC continues to work internally in Differential mode, the offset is measured at $V_{ADVREF} / 2$. The offset at V_{INP} = 0 can be computed using the transfer function and the corresponding E_G and E_O.

Figure 62-6. Gain and Offset Errors in Single-ended Mode



where:

- Full-scale error $E_{FS} = (E_{FS+}) - (E_{FS-})$, unit is LSB² code
- Offset error E_O is the offset error measured for V_{INP} = 0V
- Gain error $E_G = 100 \times E_{FS} / 2048$, unit in %

The error values in Table 62-31 include the DAC, the sample-and-hold error as well as the PGA gain error.

62.10.5 ADC Electrical Characteristics

Table 62-30. ADC INL and DNL, V_{ADVREF} = 3.3V

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
		Differential Mode				
INL	Integral Non-Linearity	–	1	–	1	LSB
DNL	Differential Non-Linearity	–	1	–	1	LSB
		Single-Ended Mode				
INL	Integral Non-Linearity	–	1.5	–	1.5	LSB
DNL	Differential Non-Linearity	–	1	–	1	LSB

Table 62-31. ADC Offset and Gain Error, V_{ADVREF} = 3.3V

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
		Differential Mode				
E_O	Differential Offset Error	–	-2.0	–	2.0	LSB
E_G	Differential Gain Error	–	-0.2	–	0.2	%
		Single-Ended Mode				
E_O	Single-ended Offset Error	–	-2.0	–	2.0	LSB
E_G	Single-ended Gain Error	–	-0.2	–	0.2	%

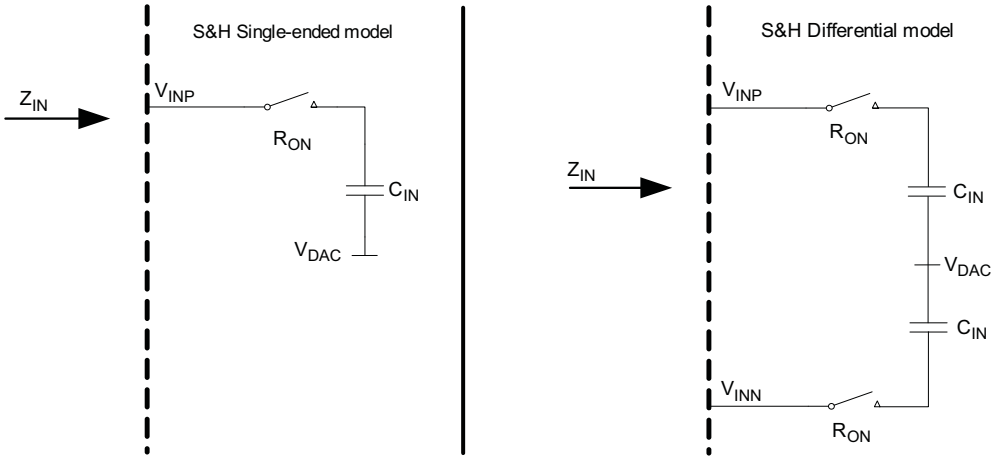
Table 62-32. ADC Analog Input Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{FS}	Analog Input Full Scale Range(1)	ADC_COR.DIFFx = 0	0	–	V_{ADVREF}	V
		ADC_COR.DIFFx = 1	$-V_{ADVREF}$	–	V_{ADVREF}	
V_{INCM}	Common Mode input range(2)	ADC_COR.DIFFx = 1	$0.4 \times V_{VDDANA}$	–	$0.6 \times V_{VDDANA}$	V
C_{IN}	ADC sampling capacitance		–	–	3	pF
C_{P_ADx}	Analog input parasitic capacitance(4)	ADx pin configured as analog input	–	–	10	
Z_{IN}	Common Mode Input impedance(3)	On ADx pin	$1 / (f_S \times C_{P\_ADx})$	–	–	Ω

Notes:
1. $V_{FS} = V_{ADx}$ in Single-ended mode, $V_{FS} = (V_{ADx} - V_{ADx+1})$ in Differential mode
2. $V_{INCM} = (V_{ADx} + V_{ADx+1}) / 2$
3. See Figure 63-7. When converting one single channel, most of the input parasitic capacitance in not switched, therefore the common mode input impedance reduces to $Z_{IN} = 1 / (f_S \times C_{IN})$
4. Includes C_{IN}

62.10.6 ADC Channel Input Impedance

Figure 62-7. Input Channel Model



where:

- Z_{IN} is the input impedance in Single-ended or Differential mode
- C_{IN} = 2 pF ±20% depending on the gain value and mode (SE or DIFF); temperature dependency is negligible
- R_{ON} is typical 2 kΩ and 8 kΩ max (worst case process and high temperature)

The following formula is used to calculate input impedance:

$$Z_{IN} = \frac{1}{f_S \times C_{IN}}$$

where:

- f_S is the sampling frequency of the ADC channel
- Typ values are used to compute ADC input impedance Z_{IN}

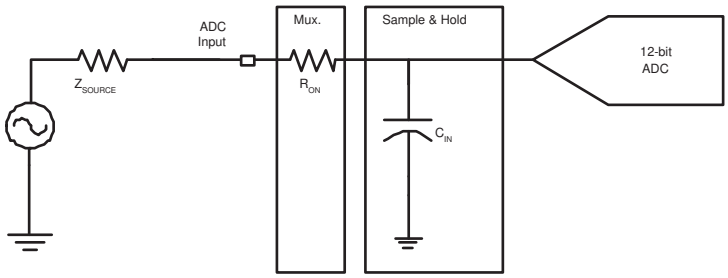
Table 62-33. Z_{IN} Input Impedance

f_S (MHz)	1	0.5	0.25	0.125	0.0625	0.03125	0.015625	0.007813
C_{IN} = 2 pF								
Z_{IN} (MΩ)	0.5	1	2	4	8	16	32	64

Track and Hold Time versus Source Output Impedance

Figure 62-8 shows a simplified acquisition path.

Figure 62-8. Simplified Acquisition Path



During the tracking phase, the ADC tracks the input signal during the tracking time shown below:

$$t_{TRACK} = n \times C_{IN} \times (R_{ON} + Z_{SOURCE}) / 1000$$

where

- Tracking time expressed in ns and Z_{SOURCE} expressed in Ω
- n = 8 for 12-bit accuracy
- R_{ON} = 2 kΩ

Table 62-34. Number of Tau:n

Resolution (bits)	12
RES	0
n	8

The ADC already includes a tracking time of 15 $t_{ADC\ Clock}$.

62.11 Analog Comparator Characteristics

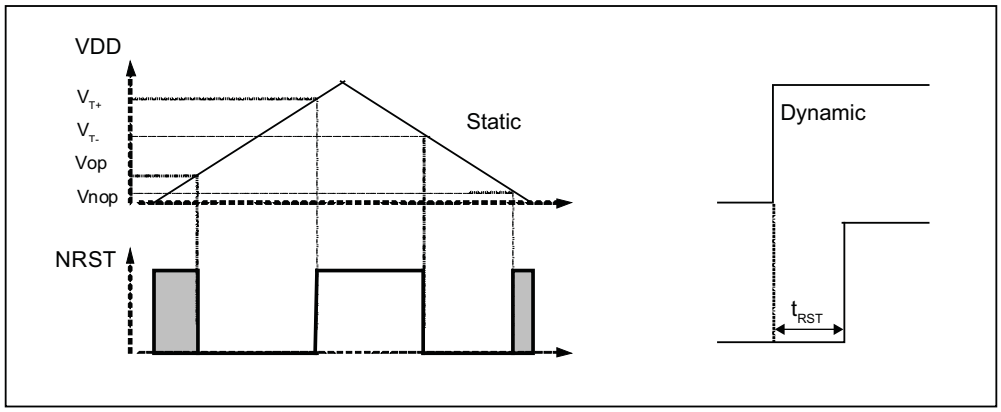
Table 62-35. Analog Comparator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDBU}	Power Supply Voltage Range (VDDBU)	Analog Comparator is supplied by VDDIN	1.62	3.3	3.6	V
V_{COMPx}	Input Voltage Range	On COMPP or COMPN input	0	–	VDDBU	V
V_{hys}	Hysteresis	–	35	–	70	mV
TPD	Propagation Delay	100mV Overdrive	–	–	350	µs
f_{in}	COMPx Input Signal Frequency	Common mode and differential	–	–	1	kHz
I_{VDDBU}	Current Consumption (VDDBU)	OFF Mode (ACC_MR.ACEN = 0)	–		50	nA
		ON Mode (ACC_MR.ACEN = 1)	–	100	200	
t_{START}	Startup Time	–	–	–	300	µs

62.12 POR Characteristics

Figure 62-9 provides a general presentation of Power-On-Reset (POR) characteristics.

Figure 62-9. General Presentation of POR Behavior



When a very slow (versus t_{RST}) supply rising slope is applied on the POR VDD pin, the reset time becomes negligible and the reset signal is released when VDD raises higher than V_{T+}.

When a very fast (versus t_{RST}) supply rising slope is applied on the POR VDD pin, the voltage threshold becomes negligible and the reset signal is released after t_{RST}. It is the smallest possible reset time.

Table 62-36. VDDBU Power-On Reset Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{T+}	Threshold Voltage Rising	–	1.3	–	1.5	V
V_{T-}	Threshold Voltage Falling	–	1.22	–	1.4	V
V_{hys}	Hysteresis Voltage	–	50	–	160	mV
t_{RST}	Reset Timeout Period	–	890	–	5100	µs

Table 62-37. VDDCORE Power-On Reset Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{T+}	Threshold Voltage Rising	–	0.927	–	1.075	V
V_{T-}	Threshold Voltage Falling	–	0.848	–	1.025	V
V_{hys}	Hysteresis Voltage	–	38	–	109	mV
t_{RST}	Reset Timeout Period	–	150	–	650	µs

Table 62-38. VDDANA Power-On Reset Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{T+}	Threshold Voltage Rising	–	1.3	–	1.5	V
V_{T-}	Threshold Voltage Falling	–	1.22	–	1.4	V
V_{hys}	Hysteresis Voltage	–	50	–	160	mV
t_{RST}	Reset Timeout Period	–	130	–	650	µs

62.13 SMC Timings

62.13.1 Timing Conditions

SMC timings are given in max corners.

Timings assuming a capacitance load on data, control and address pads are given in Table 62-39.

Table 62-39. Capacitance Load

Supply	Corner	
	Max	Min
3.3V	50 pF	5 pF
1.8V	30 pF	5 pF

In the tables that follow, t_{CPMCK} is the MCK period.

62.13.2 SMC IOSET1 Timing Extraction

62.13.2.1 SMC IOSET1 Read Timings

Table 62-40. SMC IOSET1 Read Signals - NRD Controlled (READ_MODE = 1)

Symbol	Parameter	Min		Max		Unit
	Power supply	1.8V	3.3V	1.8V	3.3V	
NO HOLD SETTINGS (nrd hold = 0)						
SMC_1	Data Setup before NRD High	16.4	15	–	–	ns
SMC_2	Data Hold after NRD High	0	0	–	–	ns
HOLD SETTINGS (nrd hold ≠ 0)						
SMC_3	Data Setup before NRD High	14.4	13	–	–	ns
SMC_4	Data Hold after NRD High	0	0	–	–	ns
HOLD or NO HOLD SETTINGS (nrd hold ≠ 0, nrd hold =0)						
SMC_5	NBS0/A0, NBS1, NBS2/A1, NBS3, A2–A25 Valid before NRD High	(nrd setup + nrd pulse) × t_{CPMCK}	(nrd setup + nrd pulse) × t_{CPMCK}	–	–	ns
SMC_6	NCS low before NRD High	(nrd setup + nrd pulse - ncs rd setup) × t_{CPMCK}	(nrd setup + nrd pulse - ncs rd setup) × t_{CPMCK}	–	–	ns
SMC_7	NRD Pulse Width	nrd pulse × t_{CPMCK}	nrd pulse × t_{CPMCK}	–	–	ns

Table 62-41. SMC IOSET1 Read Signals - NCS Controlled (READ_MODE = 0)

Symbol	Parameter	Min		Max		Unit
	Power supply	1.8V	3.3V	1.8V	3.3V	
NO HOLD SETTINGS (ncs rd hold = 0)						
SMC_8	Data Setup before NCS High	17.9	15.7	–	–	ns
SMC_9	Data Hold after NCS High	0	0	–	–	ns
HOLD SETTINGS (ncs rd hold ≠ 0)						
SMC_{10}	Data Setup before NCS High	15.9	13.7	–	–	ns
SMC_{11}	Data Hold after NCS High	0	0	–	–	ns
HOLD or NO HOLD SETTINGS (ncs rd hold ≠ 0, ncs rd hold = 0)						
SMC_{12}	NBS0/A0, NBS1, NBS2/A1, NBS3, A2–A25 valid before NCS High	(ncs rd setup + ncs rd pulse) × t_{CPMCK}	(ncs rd setup + ncs rd pulse) × t_{CPMCK}	–	–	ns
SMC_{13}	NRD low before NCS High	(ncs rd setup + ncs rd pulse - nrd setup) × t_{CPMCK}	(ncs rd setup + ncs rd pulse - nrd setup) × t_{CPMCK}	–	–	ns
SMC_{14}	NCS Pulse Width	ncs rd pulse length × t_{CPMCK}	ncs rd pulse length × t_{CPMCK}	–	–	ns

62.13.2.2 SMC IOSET1 Write Timings

Table 62-42. SMC IOSET1 Write Signals - NWE Controlled (WRITE_MODE = 1)

Symbol	Parameter	Min		Max		Unit
	Power supply	1.8V	3.3V	1.8V	3.3V	
HOLD or NO HOLD SETTINGS (nwe hold ≠ 0, nwe hold = 0)						
SMC_{15}	Data Out Valid before NWE High	nwe pulse × t_{CPMCK}	nwe pulse × t_{CPMCK}	–	–	ns
SMC_{16}	NWE Pulse Width	nwe pulse × t_{CPMCK}	nwe pulse × t_{CPMCK}	–	–	ns
SMC_{17}	NBS0/A0 NBS1, NBS2/A1, NBS3, A2–A25 valid before NWE low	nwe setup × t_{CPMCK}	nwe pulse × t_{CPMCK}	–	–	ns
SMC_{18}	NCS low before NWE high	(nwe setup - ncs rd setup + nwe pulse) × t_{CPMCK}	(nwe setup - ncs rd setup + nwe pulse) × t_{CPMCK}	–	–	ns
HOLD SETTINGS (nwe hold ≠ 0)						
SMC_{19}	NWE High to Data OUT, NBS0/A0 NBS1, NBS2/A1, NBS3, A2–A25 change	nwe hold × t_{CPMCK}	nwe hold × t_{CPMCK}	–	–	ns
SMC_{20}	NWE High to NCS Inactive(1)	(nwe hold - ncs wr hold) × t_{CPMCK}	(nwe hold - ncs wr hold) × t_{CPMCK}	–	–	ns
NO HOLD SETTINGS (nwe hold = 0)						
SMC_{21}	NWE High to Data OUT, NBS0/A0 NBS1, NBS2/A1, NBS3, A2–A25, NCS change(1)	2.3	1.3	–	–	ns

Note: 1. hold length = total cycle duration - setup duration - pulse duration. “hold length” is for “ncs wr hold length” or “NWE hold length”.

Table 62-43. SMC IOSET1 Write NCS Controlled (WRITE_MODE = 0)

Symbol	Parameter	Min		Max		Unit
	Power supply	1.8V	3.3V	1.8V	3.3V	
SMC_{22}	Data Out Valid before NCS High	ncs wr pulse × t_{CPMCK}	ncs wr pulse × t_{CPMCK}	–	–	ns
SMC_{23}	NCS Pulse Width	SMC_{14}	SMC_{14}	–	–	ns
SMC_{24}	NBS0/A0 NBS1, NBS2/A1, NBS3, A2–A25 valid before NCS low	ncs wr setup × t_{CPMCK}	ncs wr setup × t_{CPMCK}	–	–	ns
SMC_{25}	NWE low before NCS high	(ncs wr setup - nwe setup + ncs pulse) × t_{CPMCK}	(ncs wr setup - nwe setup + ncs pulse) × t_{CPMCK}	–	–	ns
SMC_{26}	NCS High to Data Out, NBS0/A0, NBS1, NBS2/A1, NBS3, A2–A25, change	ncs wr hold × t_{CPMCK}	ncs wr hold × t_{CPMCK}	–	–	ns
SMC_{27}	NCS High to NWE Inactive	(ncs wr hold - nwe hold) × t_{CPMCK}	(ncs wr hold - nwe hold) × t_{CPMCK}	–	–	ns

62.13.3 SMC IOSET2 Timing Extraction

62.13.3.1 SMC IOSET2 Read Timings

Table 62-44. SMC IOSET2 Read Signals - NRD Controlled (READ_MODE = 1)

Symbol	Parameter	Min		Max		Unit
	Power supply	1.8V	3.3V	1.8V	3.3V	
NO HOLD SETTINGS (nrd hold = 0)						
SMC_1	Data Setup before NRD High	16.5	15	–	–	ns
SMC_2	Data Hold after NRD High	0	0	–	–	ns
HOLD SETTINGS (nrd hold ≠ 0)						
SMC_3	Data Setup before NRD High	14	12.6	–	–	ns
SMC_4	Data Hold after NRD High	0	0	–	–	ns
HOLD or NO HOLD SETTINGS (nrd hold ≠ 0, nrd hold =0)						
SMC_5	NBS0/A0, NBS1, NBS2/A1, NBS3, A2–A25 Valid before NRD High	(nrd setup + nrd pulse) × t_{CPMCK}	(nrd setup + nrd pulse) × t_{CPMCK}	–	–	ns
SMC_6	NCS low before NRD High	(nrd setup + nrd pulse - ncs rd setup) × t_{CPMCK}	(nrd setup + nrd pulse - ncs rd setup) × t_{CPMCK}	–	–	ns
SMC_7	NRD Pulse Width	nrd pulse × t_{CPMCK}	nrd pulse × t_{CPMCK}	–	–	ns

Table 62-45. SMC IOSET2 Read Signals - NCS Controlled (READ_MODE = 0)

Symbol	Parameter	Min		Max		Unit
	Power supply	1.8V	3.3V	1.8V	3.3V	
NO HOLD SETTINGS (ncs rd hold = 0)						
SMC_8	Data Setup before NCS High	18.1	15.7	–	–	ns
SMC_9	Data Hold after NCS High	0	0	–	–	ns
HOLD SETTINGS (ncs rd hold ≠ 0)						
SMC_{10}	Data Setup before NCS High	15.7	13.3	–	–	ns
SMC_{11}	Data Hold after NCS High	0	0	–	–	ns
HOLD or NO HOLD SETTINGS (ncs rd hold ≠ 0, ncs rd hold = 0)						
SMC_{12}	NBS0/A0, NBS1, NBS2/A1, NBS3, A2–A25 valid before NCS High	(ncs rd setup + ncs rd pulse) × t_{CPMCK}	(ncs rd setup + ncs rd pulse) × t_{CPMCK}	–	–	ns
SMC_{13}	NRD low before NCS High	(ncs rd setup + ncs rd pulse - nrd setup) × t_{CPMCK}	(ncs rd setup + ncs rd pulse - nrd setup) × t_{CPMCK}	–	–	ns
SMC_{14}	NCS Pulse Width	ncs rd pulse length × t_{CPMCK}	ncs rd pulse length × t_{CPMCK}	–	–	ns

62.13.3.2 SMC IOSET2 Write Timings

Table 62-46. SMC IOSET2 Write Signals - NWE Controlled (WRITE_MODE = 1)

Symbol	Parameter	Min		Max		Unit
	Power supply	1.8V	3.3V	1.8V	3.3V	
HOLD or NO HOLD SETTINGS (nwe hold ≠ 0, nwe hold = 0)						
SMC_{15}	Data Out Valid before NWE High	nwe pulse × t_{CPMCK}	nwe pulse × t_{CPMCK}	–	–	ns
SMC_{16}	NWE Pulse Width	nwe pulse × t_{CPMCK}	nwe pulse × t_{CPMCK}	–	–	ns
SMC_{17}	NBS0/A0 NBS1, NBS2/A1, NBS3, A2–A25 valid before NWE low	nwe setup × t_{CPMCK}	nwe pulse × t_{CPMCK}	–	–	ns
SMC_{18}	NCS low before NWE high	(nwe setup - ncs rd setup + nwe pulse) × t_{CPMCK}	(nwe setup - ncs rd setup + nwe pulse) × t_{CPMCK}	–	–	ns
HOLD SETTINGS (nwe hold ≠ 0)						
SMC_{19}	NWE High to Data OUT, NBS0/A0 NBS1, NBS2/A1, NBS3, A2–A25 change	nwe hold × t_{CPMCK}	nwe hold × t_{CPMCK}	–	–	ns
SMC_{20}	NWE High to NCS Inactive(1)	(nwe hold - ncs wr hold) × t_{CPMCK}	(nwe hold - ncs wr hold) × t_{CPMCK}	–	–	ns
NO HOLD SETTINGS (nwe hold = 0)						
SMC_{21}	NWE High to Data OUT, NBS0/A0 NBS1, NBS2/A1, NBS3, A2–A25, NCS change((1)	1.2	0.6	–	–	ns

Note: 1. hold length = total cycle duration - setup duration - pulse duration. “hold length” is for “ncs wr hold length” or “NWE hold length”.

Table 62-47. SMC IOSET2 Write NCS Controlled (WRITE_MODE = 0)

Symbol	Parameter	Min		Max		Unit
	Power supply	1.8V	3.3V	1.8V	3.3V	
SMC_{22}	Data Out Valid before NCS High	ncs wr pulse × t_{CPMCK}	ncs wr pulse × t_{CPMCK}	–	–	ns
SMC_{23}	NCS Pulse Width	SMC_{14}	SMC_{14}	–	–	ns
SMC_{24}	NBS0/A0 NBS1, NBS2/A1, NBS3, A2–A25 valid before NCS low	ncs wr setup × t_{CPMCK}	ncs wr setup × t_{CPMCK}	–	–	ns
SMC_{25}	NWE low before NCS high	(ncs wr setup - nwe setup + ncs pulse) × t_{CPMCK}	(ncs wr setup - nwe setup + ncs pulse) × t_{CPMCK}	–	–	ns
SMC_{26}	NCS High to Data Out, NBS0/A0, NBS1, NBS2/A1, NBS3, A2–A25, change	ncs wr hold × t_{CPMCK}	ncs wr hold × t_{CPMCK}	–	–	ns
SMC_{27}	NCS High to NWE Inactive	(ncs wr hold - nwe hold) × t_{CPMCK}	(ncs wr hold - nwe hold) × t_{CPMCK}	–	–	ns

Figure 62-10. SMC Timings - NCS Controlled Read and Write

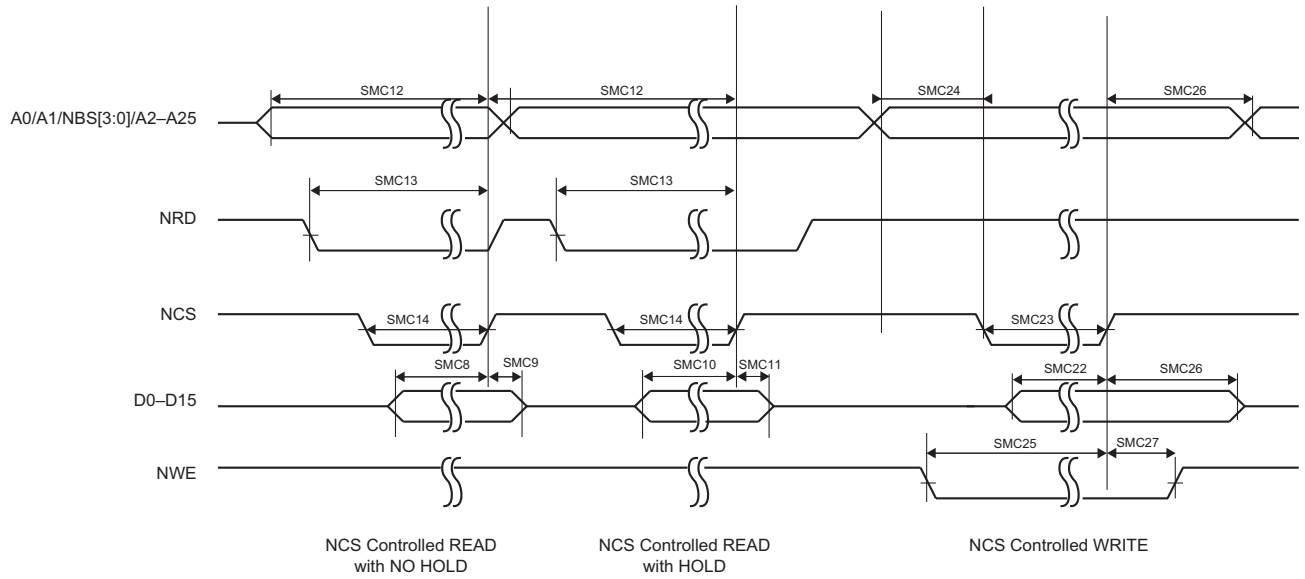
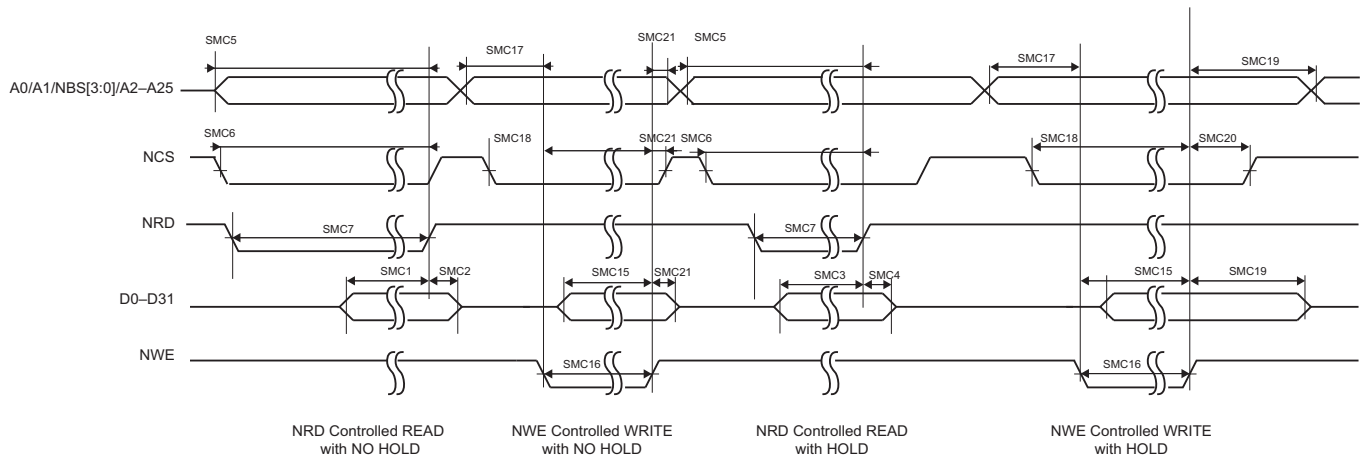


Figure 62-11. SMC Timings - NRD Controlled Read and NWE Controlled Write



62.14 SPI Timings

62.14.1 Maximum SPI Frequency

The following formulas give maximum SPI frequency in Master Read and Write modes and in Slave Read and Write modes.

Master Write Mode

The SPI is only sending data to a slave device such as an LCD, for example. The limit is given by SPI_2 (or SPI_5) timing.

Master Read Mode

$$f_{SPCK}Max = \frac{1}{SPI_0(or\ SPI_3) + t_{valid}}$$

t_{valid} is the slave time response to output data after deleting an SPCK edge. For a non-volatile memory with t_{valid} (or t_v) = 6 ns, f_{SPCK}max = 60 MHz at V_{DDIO} = 3.3V.

Slave Read Mode

In Slave mode, SPCK is the input clock for the SPI. The max SPCK frequency is given by setup and hold timings SPI_7/SPI_8 (or SPI_{10}/SPI_{11}). Since this gives a frequency well above the pad limit, the limit in Slave Read mode is given by SPCK pad.

Slave Write Mode

$$f_{SPCK}Max = \frac{1}{SPI_6(or\ SPI_9) + t_{setup}}$$

t_{setup} is the setup time from the master before sampling data (6 ns).

This gives f_{SPCK}Max = 60 MHz @ V_{DDIO} = 3.3V.

62.14.2 Timing Conditions

Timings assuming a capacitance load on MISO, SPCK and MOSI are given in Table 62-48.

Table 62-48. Capacitance Load for MISO, SPCK and MOSI (SPI0 and SPI1)

Supply	Corner	
	Max	Min
3.3V	40 pF	5 pF
1.8V	20 pF	5 pF

62.14.3 Timing Extraction

In Figure 62-13 “SPI Master Modes 1 and 2” and Figure 62-14 “SPI Master Modes 0 and 3” below, the MOSI line shifting edge is represented with a hold time = 0. However, it is important to note that for this device, the MISO line is sampled prior to the MOSI line shifting edge. As shown in Figure 62-12 “MISO Capture in Master Mode”, the device sampling point extends the propagation delay (t_p) for slave and routing delays to more than half the SPI clock period, whereas the common sampling point allows only less than half the SPI clock period.

As an example, an SPI Slave working in Mode 0 is safely driven if the SPI Master is configured in Mode 0.

Figure 62-12. MISO Capture in Master Mode

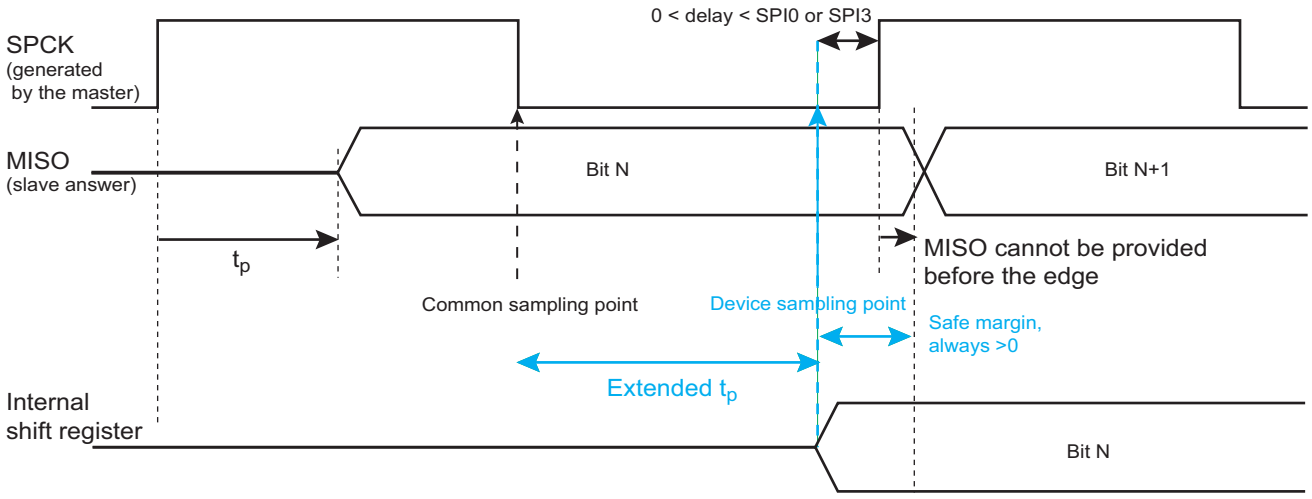


Figure 62-13. SPI Master Modes 1 and 2

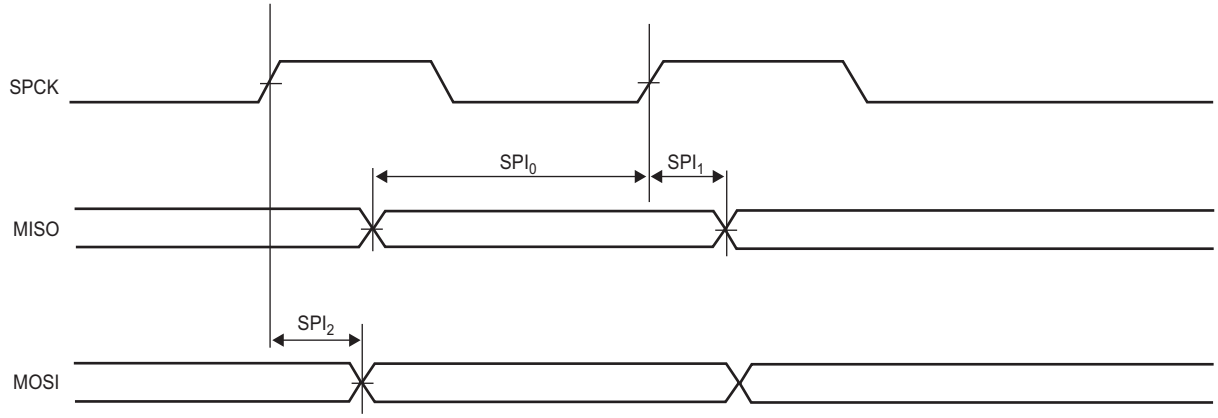


Figure 62-14. SPI Master Modes 0 and 3

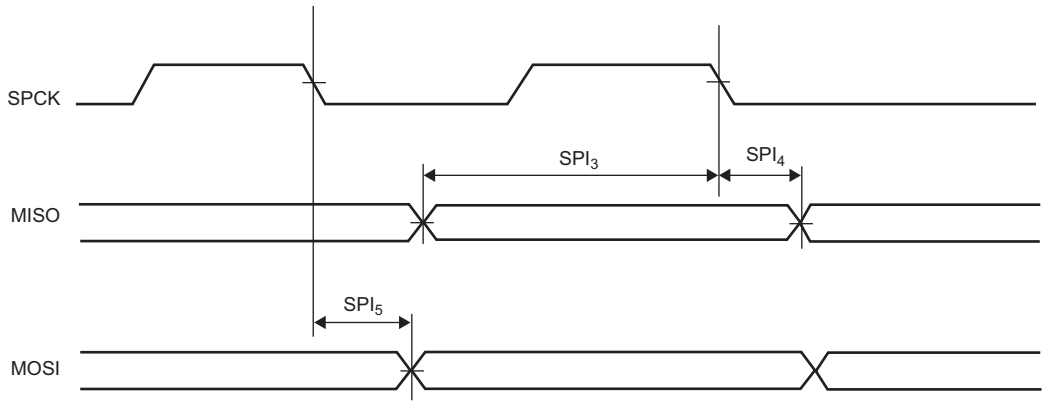


Figure 62-15. SPI Slave Modes 0 and 3

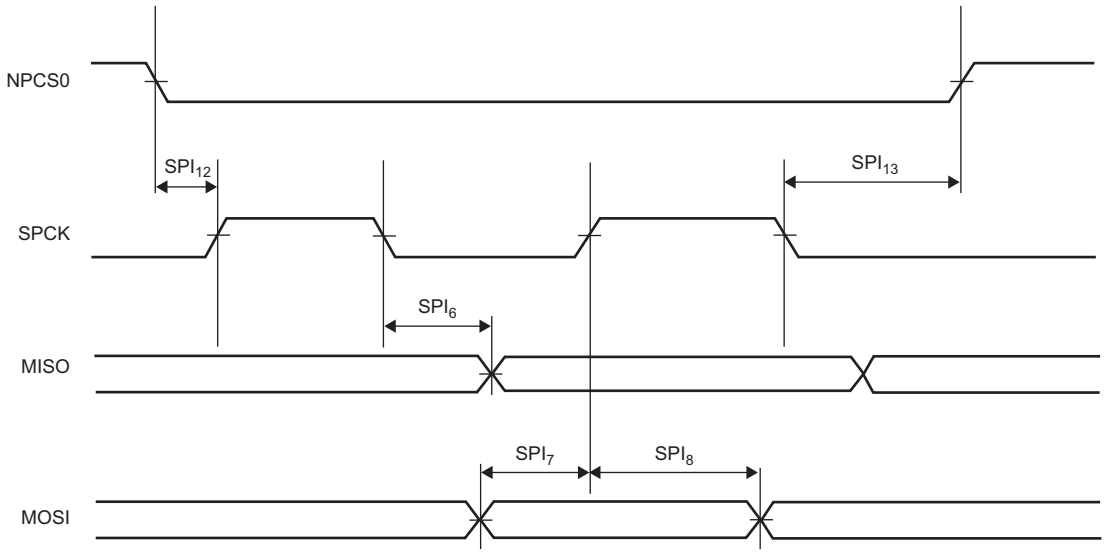


Figure 62-16. SPI Slave Modes 1 and 2

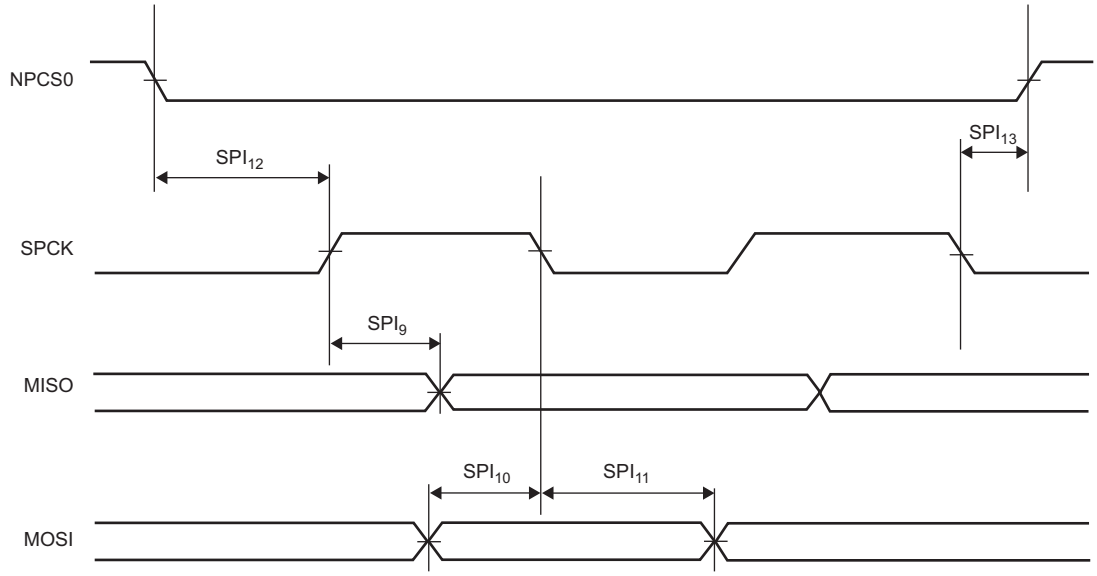


Figure 62-17. SPI Slave Mode - NPCS Timings

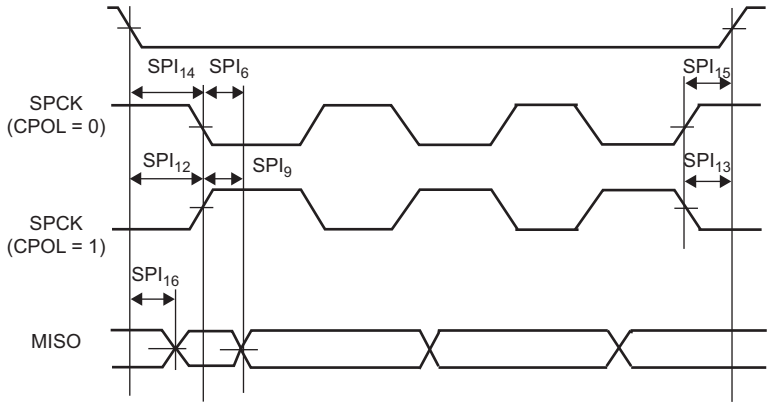


Table 62-49. SPI0 IOSET1 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master Mode					
SPI_0	MISO Setup time before SPCK rises	14.3	–	12.4	–	ns
SPI_1	MISO Hold time after SPCK rises	0	–	0	–	ns
SPI_2	SPCK rising to MOSI(1)	0	1.9	0	2.4	ns
SPI_3	MISO Setup time before SPCK falls	13.8	–	12.6	–	ns
SPI_4	MISO Hold time after SPCK falls	0	–	0	–	ns
SPI_5	SPCK falling to MOSI(1)	0	1.2	0	2.3	ns
	Slave Mode					
SPI_6	SPCK falling to MISO(1)	10.5	12.6	8.4	10.9	ns
SPI_7	MOSI Setup time before SPCK rises	1.5	–	1.4	–	ns
SPI_8	MOSI Hold time after SPCK rises	1.7	–	1.5	–	ns
SPI_9	SPCK rising to MISO(1)	10	12	8	10.2	ns
SPI_{10}	MOSI Setup time before SPCK falls	1.5	–	1.4	–	ns
SPI_{11}	MOSI Hold time after SPCK falls	1.7	–	1.5	–	ns
SPI_{12}	NPCS0 setup to SPCK rising	4.4	–	4.3	–	ns
SPI_{13}	NPCS0 hold after SPCK falling	1.5	–	1.3	–	ns
SPI_{14}	NPCS0 setup to SPCK falling	3.9	–	3.9	–	ns
SPI_{15}	NPCS0 hold after SPCK rising	0.8	–	0.5	–	ns
SPI_{16}	NPCS0 falling to MISO valid	13.3	–	11.7	–	ns

Note: 1. For output signals, the minimum and maximum access times must be extracted. The minimum access time is the time between the SPCK rising or falling edge and the signal change. The maximum access time is the time between the SPCK rising or falling edge and the signal stabilization. Figure 62-18 illustrates the minimum and maximum accesses for SPI_2. The same applies for SPI_5, SPI_6 and SPI_9.

Table 62-50. SPI0 IOSET2 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master Mode					
SPI_0	MISO Setup time before SPCK rises	14.5	–	13	–	ns
SPI_1	MISO Hold time after SPCK rises	0	–	0	–	ns
SPI_2	SPCK rising to MOSI(1)	0	2.5	0	3	ns
SPI_3	MISO Setup time before SPCK falls	14.9	–	13.6	–	ns
SPI_4	MISO Hold time after SPCK falls	0	–	0	–	ns
SPI_5	SPCK falling to MOSI(1)	0	2.7	0	3.4	ns
	Slave Mode					
SPI_6	SPCK falling to MISO(1)	10.3	12.	8.5	11.2	ns
SPI_7	MOSI Setup time before SPCK rises	2.3	–	2.2	–	ns
SPI_8	MOSI Hold time after SPCK rises	1	–	0.9	–	ns
SPI_9	SPCK rising to MISO(1)	9.9	12	8.1	10.5	ns
SPI_{10}	MOSI Setup time before SPCK falls	2.3	–	2.2	–	ns
SPI_{11}	MOSI Hold time after SPCK falls	1	–	0.9	–	ns
SPI_{12}	NPCS0 setup to SPCK rising	6.1	–	5.9	–	ns
SPI_{13}	NPCS0 hold after SPCK falling	0.8	–	0.7	–	ns
SPI_{14}	NPCS0 setup to SPCK falling	5.6	–	5.6	–	ns
SPI_{15}	NPCS0 hold after SPCK rising	0.2	–	0.1	–	ns
SPI_{16}	NPCS0 falling to MISO valid	15.1	–	13.8	–	ns

Note: 1. For output signals, the minimum and maximum access times must be extracted. The minimum access time is the time between the SPCK rising or falling edge and the signal change. The maximum access time is the time between the SPCK rising or falling edge and the signal stabilization. Figure 62-18 illustrates the minimum and maximum accesses for SPI_2. The same applies for SPI_5, SPI_6 and SPI_9.

Table 62-51. SPI1 IOSET1 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master Mode					
SPI_0	MISO Setup time before SPCK rises	14.6	–	13.1	–	ns
SPI_1	MISO Hold time after SPCK rises	0	–	0	–	ns
SPI_2	SPCK rising to MOSI(1)	0	0.8	0	1.2	ns
SPI_3	MISO Setup time before SPCK falls	15	–	13.7	–	ns
SPI_4	MISO Hold time after SPCK falls	0	–	0	–	ns
SPI_5	SPCK falling to MOSI(1)	0	0.9	0	1.6	ns
	Slave Mode					
SPI_6	SPCK falling to MISO(1)	10.3	12.4	8.3	11.2	ns
SPI_7	MOSI Setup time before SPCK rises	3.5	–	3.4	–	ns
SPI_8	MOSI Hold time after SPCK rises	0.8	–	0.7	–	ns
SPI_9	SPCK rising to MISO(1)	9.8	11.8	7.8	10.4	ns
SPI_{10}	MOSI Setup time before SPCK falls	3.5	–	3.4	–	ns
SPI_{11}	MOSI Hold time after SPCK falls	0.8	–	0.7	–	ns
SPI_{12}	NPCS0 setup to SPCK rising	4.9	–	4.8	–	ns
SPI_{13}	NPCS0 hold after SPCK falling	1.1	–	0.9	–	ns
SPI_{14}	NPCS0 setup to SPCK falling	4.4	–	4.4	–	ns
SPI_{15}	NPCS0 hold after SPCK rising	0.5	–	0.3	–	ns
SPI_{16}	NPCS0 falling to MISO valid	14.7	–	13.4	–	ns

Note: 1. For output signals, the minimum and maximum access times must be extracted. The minimum access time is the time between the SPCK rising or falling edge and the signal change. The maximum access time is the time between the SPCK rising or falling edge and the signal stabilization. Figure 62-18 illustrates the minimum and maximum accesses for SPI_2. The same applies for SPI_5, SPI_6 and SPI_9.

Table 62-52. SPI1 IOSET2 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master Mode					
SPI_0	MISO Setup time before SPCK rises	15.5	–	13.6	–	ns
SPI_1	MISO Hold time after SPCK rises	0	–	0	–	ns
SPI_2	SPCK rising to MOSI(1)	0	1.2	0	1.7	ns
SPI_3	MISO Setup time before SPCK falls	14.9	–	13.7	–	ns
SPI_4	MISO Hold time after SPCK falls	0	–	0	–	ns
SPI_5	SPCK falling to MOSI(1)	0	0.5	0	1.6	ns
	Slave Mode					
SPI_6	SPCK falling to MISO(1)	10.7	12.9	8.6	11.1	ns
SPI_7	MOSI Setup time before SPCK rises	3	–	2.9	–	ns
SPI_8	MOSI Hold time after SPCK rises	1	–	0.9	–	ns
SPI_9	SPCK rising to MISO(1)	10.3	12.4	8.2	10.5	ns
SPI_{10}	MOSI Setup time before SPCK falls	3	–	2.9	–	ns
SPI_{11}	MOSI Hold time after SPCK falls	1	–	0.9	–	ns
SPI_{12}	NPCS0 setup to SPCK rising	4.4	–	4.3	–	ns
SPI_{13}	NPCS0 hold after SPCK falling	1	–	0.9	–	ns
SPI_{14}	NPCS0 setup to SPCK falling	4	–	4	–	ns
SPI_{15}	NPCS0 hold after SPCK rising	0.4	–	0.3	–	ns
SPI_{16}	NPCS0 falling to MISO valid	14.5	–	12.9	–	ns

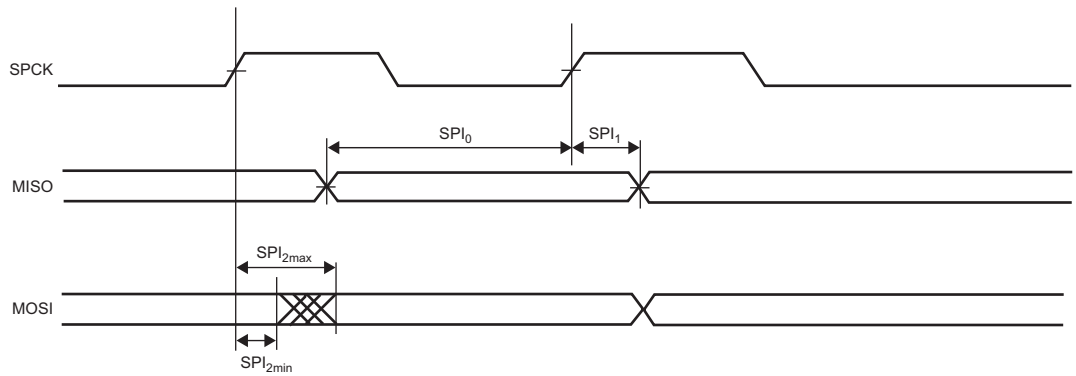
Note: 1. For output signals, the minimum and maximum access times must be extracted. The minimum access time is the time between the SPCK rising or falling edge and the signal change. The maximum access time is the time between the SPCK rising or falling edge and the signal stabilization. Figure 62-18 illustrates the minimum and maximum accesses for SPI_2. The same applies for SPI_5, SPI_6 and SPI_9.

Table 62-53. SPI1 IOSET3 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master Mode					
SPI_0	MISO Setup time before SPCK rises	13.7	–	11.7	–	ns
SPI_1	MISO Hold time after SPCK rises	0	–	0	–	ns
SPI_2	SPCK rising to MOSI(1)	0	3.1	0	2.9	ns
SPI_3	MISO Setup time before SPCK falls	14.1	–	12.4	–	ns
SPI_4	MISO Hold time after SPCK falls	0	–	0	–	ns
SPI_5	SPCK falling to MOSI(1)	0	3.1	0	3.3	ns
	Slave Mode					
SPI_6	SPCK falling to MISO(1)	9.5	11.4	7.5	9.6	ns
SPI_7	MOSI Setup time before SPCK rises	4.5	–	4.4	–	ns
SPI_8	MOSI Hold time after SPCK rises	0.7	–	0.5	–	ns
SPI_9	SPCK rising to MISO(1)	9.1	11	7.1	9.8	ns
SPI_{10}	MOSI Setup time before SPCK falls	4.5	–	4.4	–	ns
SPI_{11}	MOSI Hold time after SPCK falls	0.7	–	0.5	–	ns
SPI_{12}	NPCS0 setup to SPCK rising	5.1	–	4.9	–	ns
SPI_{13}	NPCS0 hold after SPCK falling	0.9	–	0.8	–	ns
SPI_{14}	NPCS0 setup to SPCK falling	4.7	–	4.6	–	ns
SPI_{15}	NPCS0 hold after SPCK rising	0.3	–	0.2	–	ns
SPI_{16}	NPCS0 falling to MISO valid	13.9	–	12.2	–	ns

Note: 1. For output signals, the minimum and maximum access times must be extracted. The minimum access time is the time between the SPCK rising or falling edge and the signal change. The maximum access time is the time between the SPCK rising or falling edge and the signal stabilization. Figure 62-18 illustrates the minimum and maximum accesses for SPI_2. The same applies for SPI_5, SPI_6 and SPI_9.

Figure 62-18. Minimum and Maximum Access Time for SPI Output Signal



62.15 FLEXCOM Timings

62.15.1 Timing Conditions

Timings assuming a capacitance load on MISO, SPCK and MOSI are given in Table 62-48.

Table 62-54. Capacitance Load for MISO, SPCK and MOSI (FLEXCOM 0, 1, 2, 3, 4)

Supply	Corner	
	Max	Min
3.3V	40 pF	5 pF
1.8V	20 pF	5 pF

62.15.2 Timing Extraction

Figure 62-19. FLEXCOM in SPI Master Modes 1 and 2

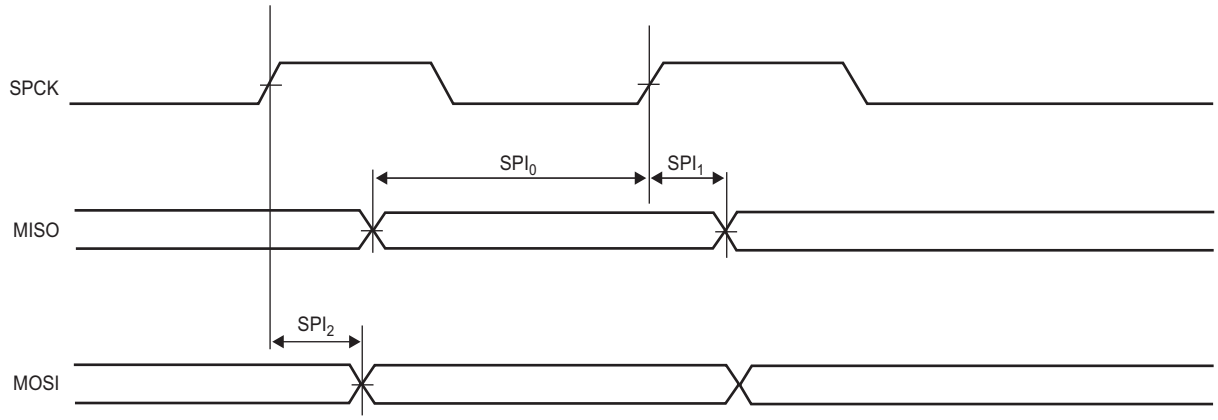


Figure 62-20. FLEXCOM in SPI Master Modes 0 and 3

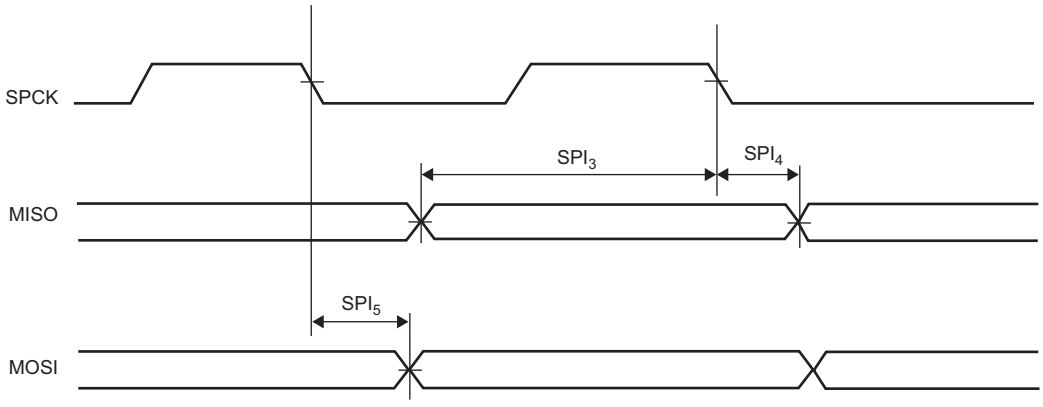


Figure 62-21. FLEXCOM in SPI Slave Modes 0 and 3

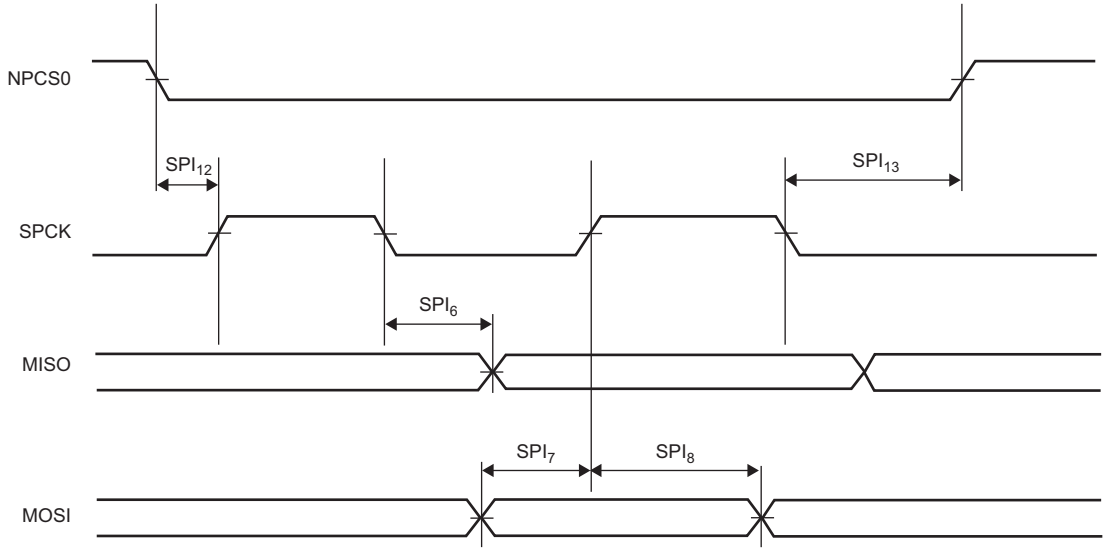


Figure 62-22. FLEXCOM in SPI Slave Modes 1 and 2

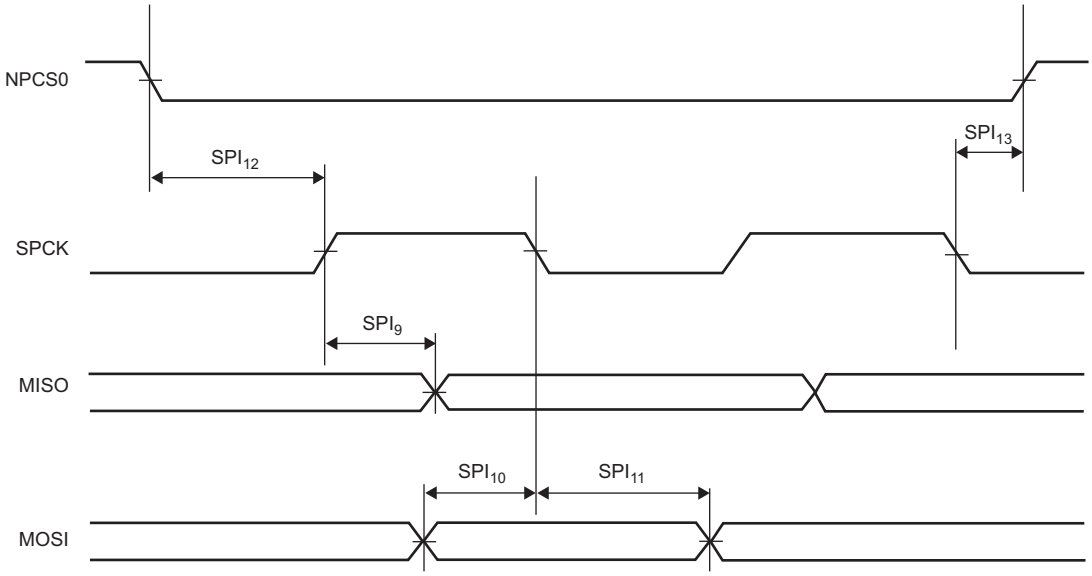


Figure 62-23. FLEXCOM in SPI Slave Mode - NPCS Timings

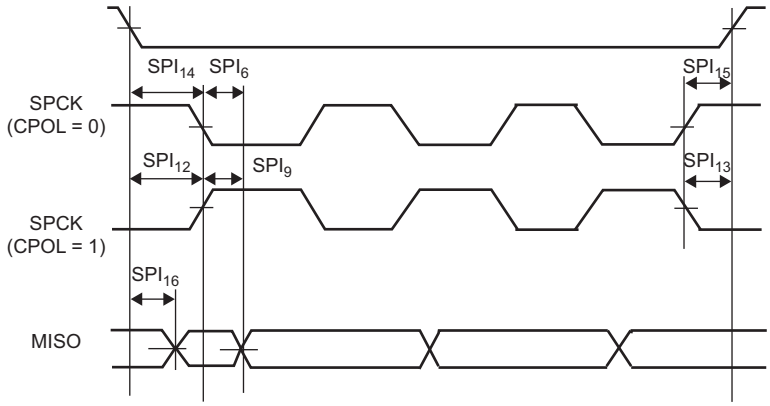


Table 62-55. FLEXCOM0 in SPI Mode IOSET1 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master Mode					
SPI_0	MISO Setup time before SPCK rises	14.3	–	12.8	–	ns
SPI_1	MISO Hold time after SPCK rises	0	–	0	–	ns
SPI_2	SPCK rising to MOSI(1)	0	3.9	0	4.2	ns
SPI_3	MISO Setup time before SPCK falls	14.7	–	13.4	–	ns
SPI_4	MISO Hold time after SPCK falls	0	–	0	–	ns
SPI_5	SPCK falling to MOSI(1)	0	3.9	0	4.6	ns
	Slave Mode					
SPI_6	SPCK falling to MISO(1)	10.9	13.4	9.1	11.9	ns
SPI_7	MOSI Setup time before SPCK rises	5.3	–	5	–	ns
SPI_8	MOSI Hold time after SPCK rises	0.4	–	0.3	–	ns
SPI_9	SPCK rising to MISO(1)	10.7	13.1	9	11.5	ns
SPI_{10}	MOSI Setup time before SPCK falls	5.3	–	5	–	ns
SPI_{11}	MOSI Hold time after SPCK falls	0.4	–	0.3	–	ns
SPI_{12}	NPCS0 setup to SPCK rising	5.6	–	5.4	–	ns
SPI_{13}	NPCS0 hold after SPCK falling	0.7	–	0.6	–	ns
SPI_{14}	NPCS0 setup to SPCK falling	5.4	–	5.2	–	ns
SPI_{15}	NPCS0 hold after SPCK rising	0.2	–	0.1	–	ns
SPI_{16}	NPCS0 falling to MISO valid	17.5	–	16.2	–	ns

Note: 1. For output signals, the minimum and maximum access times must be extracted. The minimum access time is the time between the SPCK rising or falling edge and the signal change. The maximum access time is the time between the SPCK rising or falling edge and the signal stabilization. Figure 62-24 illustrates the minimum and maximum accesses for SPI_2. The same applies for SPI_5, SPI_6 and SPI_9.

Table 62-56. FLEXCOM1 in SPI Mode IOSET1 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
Master Mode						
SPI_0	MISO Setup time before SPCK rises	15.7	–	13.8	–	ns
SPI_1	MISO Hold time after SPCK rises	0	–	0	–	ns
SPI_2	SPCK rising to MOSI(1)	0	2.5	0	2.6	ns
SPI_3	MISO Setup time before SPCK falls	15.2	–	13.9	–	ns
SPI_4	MISO Hold time after SPCK falls	0	–	0	–	ns
SPI_5	SPCK falling to MOSI(1)	0	1.7	0	2.4	ns
Slave Mode						
SPI_6	SPCK falling to MISO(1)	11.4	13.7	9.4	12.3	ns
SPI_7	MOSI Setup time before SPCK rises	2.4	–	2.1	–	ns
SPI_8	MOSI Hold time after SPCK rises	1	–	0.9	–	ns
SPI_9	SPCK rising to MISO(1)	11	13.1	9	11.6	ns
SPI_{10}	MOSI Setup time before SPCK falls	2.4	–	2.1	–	ns
SPI_{11}	MOSI Hold time after SPCK falls	1	–	0.9	–	ns
SPI_{12}	NPCS0 setup to SPCK rising	3.9	–	3.7	–	ns
SPI_{13}	NPCS0 hold after SPCK falling	1.1	–	1	–	ns
SPI_{14}	NPCS0 setup to SPCK falling	3.4	–	3.3	–	ns
SPI_{15}	NPCS0 hold after SPCK rising	0.6	–	0.4	–	ns
SPI_{16}	NPCS0 falling to MISO valid	16.8	–	15.5	–	ns

Note: 1. For output signals, the minimum and maximum access times must be extracted. The minimum access time is the time between the SPCK rising or falling edge and the signal change. The maximum access time is the time between the SPCK rising or falling edge and the signal stabilization. Figure 62-24 illustrates the minimum and maximum accesses for SPI_2. The same applies for SPI_5, SPI_6 and SPI_9.

Table 62-57. FLEXCOM2 in SPI Mode IOSET1 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master Mode					
SPI_0	MISO Setup time before SPCK rises	15.3	–	13.4	–	ns
SPI_1	MISO Hold time after SPCK rises	0	–	0	–	ns
SPI_2	SPCK rising to MOSI(1)	0	2.5	0	3	ns
SPI_3	MISO Setup time before SPCK falls	15.6	–	13.7	–	ns
SPI_4	MISO Hold time after SPCK falls	0	–	0	–	ns
SPI_5	SPCK falling to MOSI(1)	0	2.6	0	3.1	ns
	Slave Mode					
SPI_6	SPCK falling to MISO(1)	11.7	13.5	9.4	11.7	ns
SPI_7	MOSI Setup time before SPCK rises	2	–	1.6	–	ns
SPI_8	MOSI Hold time after SPCK rises	0.5	–	0.5	–	ns
SPI_9	SPCK rising to MISO(1)	11.7	13.5	9.4	11.5	ns
SPI_{10}	MOSI Setup time before SPCK falls	2	–	1.6	–	ns
SPI_{11}	MOSI Hold time after SPCK falls	0.5	–	0.5	–	ns
SPI_{12}	NPCS0 setup to SPCK rising	4	–	3.7	–	ns
SPI_{13}	NPCS0 hold after SPCK falling	0.6	–	0.6	–	ns
SPI_{14}	NPCS0 setup to SPCK falling	3.9	–	3.7	–	ns
SPI_{15}	NPCS0 hold after SPCK rising	0.4	–	0.3	–	ns
SPI_{16}	NPCS0 falling to MISO valid	16.8	–	13.6	–	ns

Note: 1. For output signals, the minimum and maximum access times must be extracted. The minimum access time is the time between the SPCK rising or falling edge and the signal change. The maximum access time is the time between the SPCK rising or falling edge and the signal stabilization. Figure 62-24 illustrates the minimum and maximum accesses for SPI_2. The same applies for SPI_5, SPI_6 and SPI_9.

Table 62-58. FLEXCOM2 in SPI Mode IOSET2 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master Mode					
SPI_0	MISO Setup time before SPCK rises	13.3	–	11.2	–	ns
SPI_1	MISO Hold time after SPCK rises	0	–	0	–	ns
SPI_2	SPCK rising to MOSI(1)	0	4.4	0	4.1	ns
SPI_3	MISO Setup time before SPCK falls	4.3	–	12.5	–	ns
SPI_4	MISO Hold time after SPCK falls	0	–	0	–	ns
SPI_5	SPCK falling to MOSI(1)	0.3	4.4	0.4	4.4	ns
	Slave Mode					
SPI_6	SPCK falling to MISO(1)	10.1	12.2	8.2	10.4	ns
SPI_7	MOSI Setup time before SPCK rises	3.3	–	3.1	–	ns
SPI_8	MOSI Hold time after SPCK rises	0.8	–	0.7	–	ns
SPI_9	SPCK rising to MISO(1)	9.8	11.8	7.9	9.8	ns
SPI_{10}	MOSI Setup time before SPCK falls	3.3	–	3.1	–	ns
SPI_{11}	MOSI Hold time after SPCK falls	0.8	–	0.7	–	ns
SPI_{12}	NPCS0 setup to SPCK rising	5.3	–	5.2	–	ns
SPI_{13}	NPCS0 hold after SPCK falling	0.8	–	0.6	–	ns
SPI_{14}	NPCS0 setup to SPCK falling	5	–	4.9	–	ns
SPI_{15}	NPCS0 hold after SPCK rising	0.2	–	0.1	–	ns
SPI_{16}	NPCS0 falling to MISO valid	15.9	–	14.2	–	ns

Note: 1. For output signals, the minimum and maximum access times must be extracted. The minimum access time is the time between the SPCK rising or falling edge and the signal change. The maximum access time is the time between the SPCK rising or falling edge and the signal stabilization. Figure 62-24 illustrates the minimum and maximum accesses for SPI_2. The same applies for SPI_5, SPI_6 and SPI_9.

Table 62-59. FLEXCOM3 in SPI Mode IOSET1 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master Mode					
SPI_0	MISO Setup time before SPCK rises	14.6	–	12.7	–	ns
SPI_1	MISO Hold time after SPCK rises	0	–	0	–	ns
SPI_2	SPCK rising to MOSI(1)	0	2.6	0	2.9	ns
SPI_3	MISO Setup time before SPCK falls	14.2	–	13	–	ns
SPI_4	MISO Hold time after SPCK falls	0	–	0	–	ns
SPI_5	SPCK falling to MOSI(1)	0	1.8	0	2.9	ns
	Slave Mode					
SPI_6	SPCK falling to MISO(1)	11.6	14.1	9.5	12.7	ns
SPI_7	MOSI Setup time before SPCK rises	3.3	–	3.2	–	ns
SPI_8	MOSI Hold time after SPCK rises	0.9	–	0.7	–	ns
SPI_9	SPCK rising to MISO(1)	11.2	13.6	9.1	12.2	ns
SPI_{10}	MOSI Setup time before SPCK falls	3.3	–	3.2	–	ns
SPI_{11}	MOSI Hold time after SPCK falls	0.9	–	0.7	–	ns
SPI_{12}	NPCS0 setup to SPCK rising	2.8	–	2.6	–	ns
SPI_{13}	NPCS0 hold after SPCK falling	1.5	–	1.3	–	ns
SPI_{14}	NPCS0 setup to SPCK falling	2.3	–	2.2	–	ns
SPI_{15}	NPCS0 hold after SPCK rising	1	–	0.7	–	ns
SPI_{16}	NPCS0 falling to MISO valid	15.4	–	14.1	–	ns

Note: 1. For output signals, the minimum and maximum access times must be extracted. The minimum access time is the time between the SPCK rising or falling edge and the signal change. The maximum access time is the time between the SPCK rising or falling edge and the signal stabilization. Figure 62-24 illustrates the minimum and maximum accesses for SPI_2. The same applies for SPI_5, SPI_6 and SPI_9.

Table 62-60. FLEXCOM3 in SPI Mode IOSET2 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master Mode					
SPI_0	MISO Setup time before SPCK rises	14.1	–	12.6	–	ns
SPI_1	MISO Hold time after SPCK rises	0	–	0	–	ns
SPI_2	SPCK rising to MOSI(1)	0	3.8	0	4.1	ns
SPI_3	MISO Setup time before SPCK falls	14.9	–	13.7	–	ns
SPI_4	MISO Hold time after SPCK falls	0	–	0	–	ns
SPI_5	SPCK falling to MOSI(1)	0	3.9	0	4.5	ns
	Slave Mode					
SPI_6	SPCK falling to MISO(1)	10.6	13.1	8.6	11.8	ns
SPI_7	MOSI Setup time before SPCK rises	3.7	–	3.5	–	ns
SPI_8	MOSI Hold time after SPCK rises	0.6	–	0.5	–	ns
SPI_9	SPCK rising to MISO(1)	10.2	12.6	8.2	11.1	ns
SPI_{10}	MOSI Setup time before SPCK falls	3.7	–	3.5	–	ns
SPI_{11}	MOSI Hold time after SPCK falls	0.6	–	0.5	–	ns
SPI_{12}	NPCS0 setup to SPCK rising	4.5	–	4.3	–	ns
SPI_{13}	NPCS0 hold after SPCK falling	1	–	0.9	–	ns
SPI_{14}	NPCS0 setup to SPCK falling	4	–	3.9	–	ns
SPI_{15}	NPCS0 hold after SPCK rising	0.4	–	0.3	–	ns
SPI_{16}	NPCS0 falling to MISO valid	16.9	–	15.6	–	ns

Note: 1. For output signals, the minimum and maximum access times must be extracted. The minimum access time is the time between the SPCK rising or falling edge and the signal change. The maximum access time is the time between the SPCK rising or falling edge and the signal stabilization. Figure 62-24 illustrates the minimum and maximum accesses for SPI_2. The same applies for SPI_5, SPI_6 and SPI_9.

Table 62-61. FLEXCOM3 in SPI Mode IOSET3 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master Mode					
SPI_0	MISO Setup time before SPCK rises	14.2	–	12.7	–	ns
SPI_1	MISO Hold time after SPCK rises	0	–	0	–	ns
SPI_2	SPCK rising to MOSI(1)	0	3.4	0	3.7	ns
SPI_3	MISO Setup time before SPCK falls	15.1	–	13.8	–	ns
SPI_4	MISO Hold time after SPCK falls	0	–	0	–	ns
SPI_5	SPCK falling to MOSI(1)(1)	0	3.5	0	4.2	ns
	Slave Mode					
SPI_6	SPCK falling to MISO(1)	11.4	14.1	9.4	12.8	ns
SPI_7	MOSI Setup time before SPCK rises	5.4	–	5.1	–	ns
SPI_8	MOSI Hold time after SPCK rises	0.4	–	0.3	–	ns
SPI_9	SPCK rising to MISO(1)	11	13.6	9	12.2	ns
SPI_{10}	MOSI Setup time before SPCK falls	5.4	–	5.1	–	ns
SPI_{11}	MOSI Hold time after SPCK falls	0.4	–	0.3	–	ns
SPI_{12}	NPCS0 setup to SPCK rising	4.3	–	4.2	–	ns
SPI_{13}	NPCS0 hold after SPCK falling	1.1	–	0.9	–	ns
SPI_{14}	NPCS0 setup to SPCK falling	3.8	–	3.7	–	ns
SPI_{15}	NPCS0 hold after SPCK rising	0.5	–	0.3	–	ns
SPI_{16}	NPCS0 falling to MISO valid	17.5	–	16.2	–	ns

Note: 1. For output signals, the minimum and maximum access times must be extracted. The minimum access time is the time between the SPCK rising or falling edge and the signal change. The maximum access time is the time between the SPCK rising or falling edge and the signal stabilization. Figure 62-24 illustrates the minimum and maximum accesses for SPI_2. The same applies for SPI_5, SPI_6 and SPI_9.

Table 62-62. FLEXCOM4 in SPI Mode IOSET1 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master Mode					
SPI_0	MISO Setup time before SPCK rises	14.7	–	13.1	–	ns
SPI_1	MISO Hold time after SPCK rises	0	–	0	–	ns
SPI_2	SPCK rising to MOSI(1)	0	2.7	0	3.2	ns
SPI_3	MISO Setup time before SPCK falls	15.2	–	13.8	–	ns
SPI_4	MISO Hold time after SPCK falls	0	–	0	–	ns
SPI_5	SPCK falling to MOSI(1)	0	2.9	0	3.6	ns
	Slave Mode					
SPI_6	SPCK falling to MISO(1)	10.9	13.2	8.9	11.9	ns
SPI_7	MOSI Setup time before SPCK rises	3.3	–	3.2	–	ns
SPI_8	MOSI Hold time after SPCK rises	0.7	–	0.6	–	ns
SPI_9	SPCK rising to MISO(1)	10.5	12.7	8.5	11.3	ns
SPI_{10}	MOSI Setup time before SPCK falls	3.3	–	3.2	–	ns
SPI_{11}	MOSI Hold time after SPCK falls	0.7	–	0.6	–	ns
SPI_{12}	NPCS0 setup to SPCK rising	5.7	–	5.5	–	ns
SPI_{13}	NPCS0 hold after SPCK falling	0.6	–	0.5	–	ns
SPI_{14}	NPCS0 setup to SPCK falling	5.2	–	5	–	ns
SPI_{15}	NPCS0 hold after SPCK rising	0	–	0	–	ns
SPI_{16}	NPCS0 falling to MISO valid	17.8	–	16.5	–	ns

Note: 1. For output signals, the minimum and maximum access times must be extracted. The minimum access time is the time between the SPCK rising or falling edge and the signal change. The maximum access time is the time between the SPCK rising or falling edge and the signal stabilization. Figure 62-24 illustrates the minimum and maximum accesses for SPI_2. The same applies for SPI_5, SPI_6 and SPI_9.

Table 62-63. FLEXCOM4 in SPI Mode IOSET2 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master Mode					
SPI_0	MISO Setup time before SPCK rises	14.7	–	11.5	–	ns
SPI_1	MISO Hold time after SPCK rises	0	–	0	–	ns
SPI_2	SPCK rising to MOSI(1)	0	2.7	0	4.7	ns
SPI_3	MISO Setup time before SPCK falls	15.2	–	12.8	–	ns
SPI_4	MISO Hold time after SPCK falls	0	–	0	–	ns
SPI_5	SPCK falling to MOSI(1)	0	2.9	0.8	5.1	ns
	Slave Mode					
SPI_6	SPCK falling to MISO(1)	10.9	13.2	7.8	10.2	ns
SPI_7	MOSI Setup time before SPCK rises	3.3	–	2.2	–	ns
SPI_8	MOSI Hold time after SPCK rises	0.7	–	0.8	–	ns
SPI_9	SPCK rising to MISO(1)	10.5	12.7	7.7	9.9	ns
SPI_{10}	MOSI Setup time before SPCK falls	3.3	–	2.2	–	ns
SPI_{11}	MOSI Hold time after SPCK falls	0.7	–	0.8	–	ns
SPI_{12}	NPCS0 setup to SPCK rising	5.6	–	3.3	–	ns
SPI_{13}	NPCS0 hold after SPCK falling	0.6	–	1.1	–	ns
SPI_{14}	NPCS0 setup to SPCK falling	5.2	–	3.1	–	ns
SPI_{15}	NPCS0 hold after SPCK rising	0	–	0.8	–	ns
SPI_{16}	NPCS0 falling to MISO valid	17.8	–	12.9	–	ns

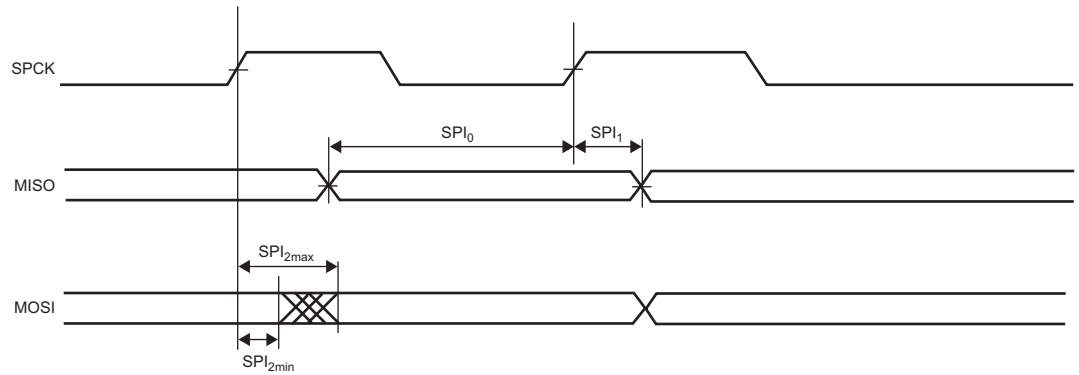
Note: 1. For output signals, the minimum and maximum access times must be extracted. The minimum access time is the time between the SPCK rising or falling edge and the signal change. The maximum access time is the time between the SPCK rising or falling edge and the signal stabilization. Figure 62-24 illustrates the minimum and maximum accesses for SPI_2. The same applies for SPI_5, SPI_6 and SPI_9.

Table 62-64. FLEXCOM4 in SPI Mode IOSET3 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master Mode					
SPI_0	MISO Setup time before SPCK rises	14.2	–	12.2	–	ns
SPI_1	MISO Hold time after SPCK rises	0	–	0	–	ns
SPI_2	SPCK rising to MOSI(1)	0	3.6	0	3.6	ns
SPI_3	MISO Setup time before SPCK falls	15.1	–	13.3	–	ns
SPI_4	MISO Hold time after SPCK falls	0	–	0	–	ns
SPI_5	SPCK falling to MOSI(1)	0.1	3.7	0.1	4	ns
	Slave Mode					
SPI_6	SPCK falling to MISO(1)	9.9	12.4	8	10.5	ns
SPI_7	MOSI Setup time before SPCK rises	3.9	–	3.7	–	ns
SPI_8	MOSI Hold time after SPCK rises	0.8	–	0.7	–	ns
SPI_9	SPCK rising to MISO(1)	9.5	11.9	7.7	9.9	ns
SPI_{10}	MOSI Setup time before SPCK falls	3.9	–	3.7	–	ns
SPI_{11}	MOSI Hold time after SPCK falls	0.8	–	0.7	–	ns
SPI_{12}	NPCS0 setup to SPCK rising	4.8	–	4.6	–	ns
SPI_{13}	NPCS0 hold after SPCK falling	1	–	0.9	–	ns
SPI_{14}	NPCS0 setup to SPCK falling	4.4	–	4.3	–	ns
SPI_{15}	NPCS0 hold after SPCK rising	0.5	–	0.3	–	ns
SPI_{16}	NPCS0 falling to MISO valid	16	–	14.2	–	ns

Note: 1. For output signals, the minimum and maximum access times must be extracted. The minimum access time is the time between the SPCK rising or falling edge and the signal change. The maximum access time is the time between the SPCK rising or falling edge and the signal stabilization. Figure 62-24 illustrates the minimum and maximum accesses for SPI_2. The same applies for SPI_5, SPI_6 and SPI_9.

Figure 62-24. Minimum and Maximum Access Time for SPI Output Signal



62.16 QSPI Timings

62.16.1 Maximum QSPI Frequency

The following formulas give maximum QSPI frequency in Master read and write modes.

Master Write Mode

The QSPI sends data to a slave device only, e.g. an LCD. The limit is given by $QSPI_2$ (or $QSPI_5$) timing.

Master Read Mode

$$f_{SCK}\text{max} = \frac{1}{QSPI_0(\text{or } QSPI_3) + t_{VALID}}$$

t_{VALID} is the slave time response to output data after detecting a QSCK edge.

For a non-volatile memory with t_{VALID} (or t_v) = 12 ns, f_{SCK}max = 67 MHz at V_{DDIO} = 3.3V.

62.16.2 Timing Conditions

Timings assuming a capacitance load are given in Table 62-65.

Table 62-65. Capacitance Load (QSPI 0 and QSPI1)

Supply	Corner	
	Max	Min
3.3V	30 pF	5 pF
1.8V	20 pF	5 pF

62.16.3 Timing Extraction

Figure 62-25. QSPI Master Mode 0

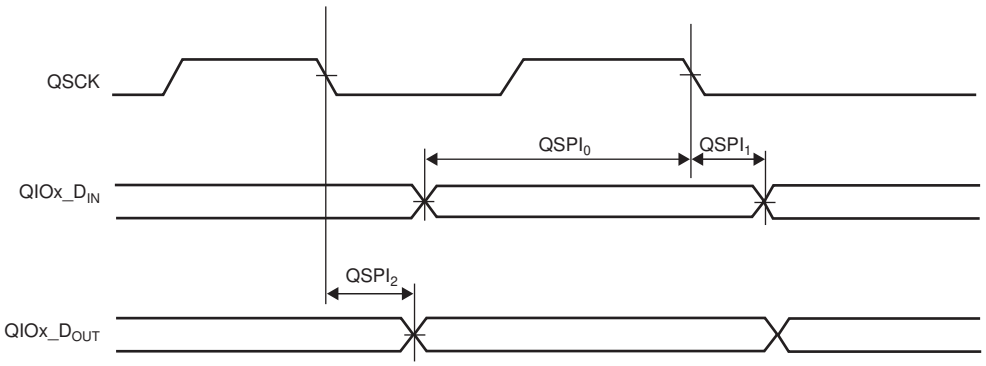


Figure 62-26. QSPI Master Mode 1

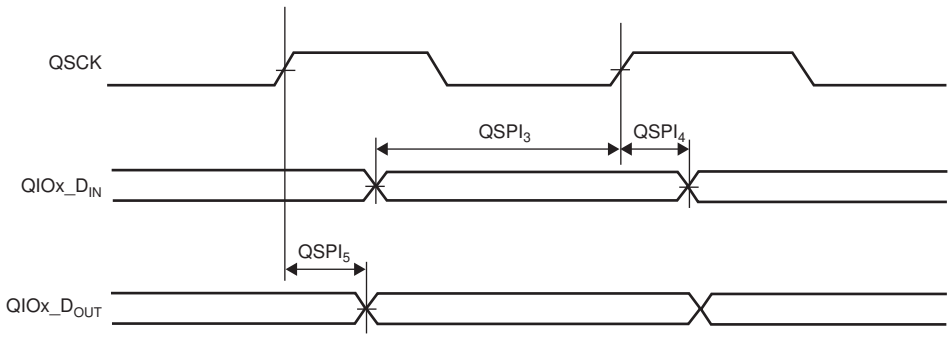


Figure 62-27. QSPI Master Mode 2

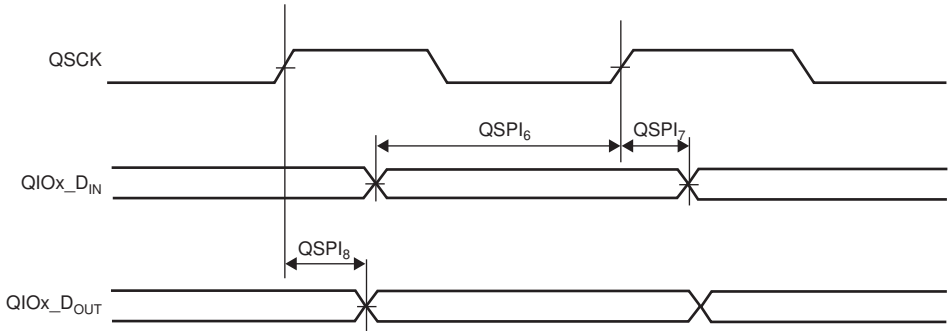


Figure 62-28. QSPI Master Mode 3

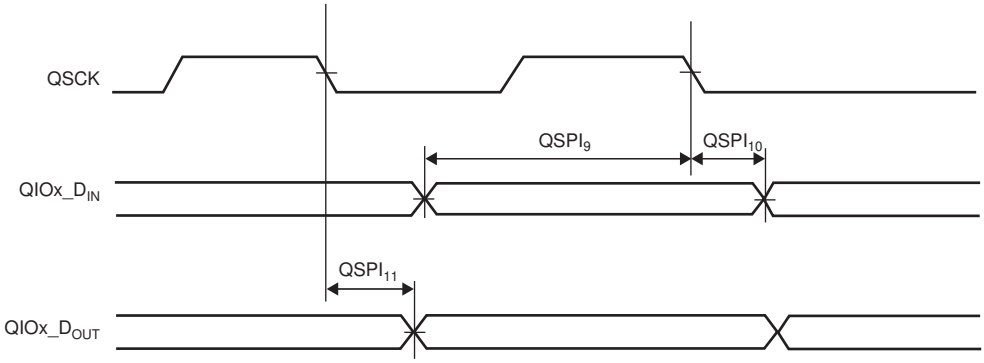


Table 62-66. QSPI0 IOSET1 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
Mode 0						
$QSPI_0$	QIOx Input setup time before SCK falls	1.5	–	1.4	–	ns
$QSPI_1$	QIOx Input hold time after SCK falls	0.4	–	0.5	–	ns
$QSPI_2$	SCK falling to QIOx valid	0	3.4	0	2.4	ns
Mode 1						
$QSPI_3$	QIOx Input setup time before SCK rises	11	–	8.7	–	ns
$QSPI_4$	QIOx Input hold time after SCK rises	0.1	–	0	–	ns
$QSPI_5$	SCK rising to QIOx valid	0	3	0	2.2	ns
Mode 2						
$QSPI_6$	QIOx Input setup time before SCK rises	1.7	–	1.4	–	ns
$QSPI_7$	QIOx Input hold time after SCK rises	0.1	–	0	–	ns
$QSPI_8$	SCK rising to QIOx valid	0	3.3	0	2.3	ns
Mode 3						
$QSPI_9$	QIOx Input setup time before SCK falls	11	–	8.9	–	ns
$QSPI_{10}$	QIOx Input hold time after SCK falls	0.4	–	0.4	–	ns
$QSPI_{11}$	SCK falling to QIOx valid	0	3.2	0	2.4	ns

Table 62-67. QSPI0 IOSET2 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
Mode 0						
$QSPI_0$	QIOx Input setup time before SCK falls	2.2	–	2.2	–	ns
$QSPI_1$	QIOx Input hold time after SCK falls	0.8	–	0.7	–	ns
$QSPI_2$	SCK falling to QIOx valid	0	1.8	0	2	ns
Mode 1						
$QSPI_3$	QIOx Input setup time before SCK rises	12.7	–	10.5	–	ns
$QSPI_4$	QIOx Input hold time after SCK rises	0.3	–	0.2	–	ns
$QSPI_5$	SCK rising to QIOx valid	0	2.2	0	1.9	ns
Mode 2						
$QSPI_6$	QIOx Input setup time before SCK rises	2.6	–	2.5	–	ns
$QSPI_7$	QIOx Input hold time after SCK rises	0.3	–	0.2	–	ns
$QSPI_8$	SCK rising to QIOx valid	0	2.4	0	2	ns
Mode 3						
$QSPI_9$	QIOx Input setup time before SCK falls	12	–	10.5	–	ns
$QSPI_{10}$	QIOx Input hold time after SCK falls	0.8	–	0.7	–	ns
$QSPI_{11}$	SCK falling to QIOx valid	0	1.5	0	1.9	ns

Table 62-68. QSPI0 IOSET3 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
Mode 0						
$QSPI_0$	QIOx Input setup time before SCK falls	1.8	–	1.7	–	ns
$QSPI_1$	QIOx Input hold time after SCK falls	0.8	–	0.7	–	ns
$QSPI_2$	SCK falling to QIOx valid	0	1.8	0	2.1	ns
Mode 1						
$QSPI_3$	QIOx Input setup time before SCK rises	12.3	–	10.1	–	ns
$QSPI_4$	QIOx Input hold time after SCK rises	0.5	–	0.3	–	ns
$QSPI_5$	SCK rising to QIOx valid	0	2.2	0	2	ns
Mode 2						
$QSPI_6$	QIOx Input setup time before SCK rises	2	–	1.7	–	ns
$QSPI_7$	QIOx Input hold time after SCK rises	0.5	–	0.3	–	ns
$QSPI_8$	SCK rising to QIOx valid	0	2.5	0	2.2	ns
Mode 3						
$QSPI_9$	QIOx Input setup time before SCK falls	11.7	–	10.2	–	ns
$QSPI_{10}$	QIOx Input hold time after SCK falls	0.8	–	0.7	–	ns
$QSPI_{11}$	SCK falling to QIOx valid	0	1.5	1.2	2	ns

Table 62-69. QSPI1 IOSET1 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
Mode 0						
$QSPI_0$	QIOx Input setup time before SCK falls	1.1	–	0.9	–	ns
$QSPI_1$	QIOx Input hold time after SCK falls	1	–	0.7	–	ns
$QSPI_2$	SCK falling to QIOx valid	0	3.2	0	2.4	ns
Mode 1						
$QSPI_3$	QIOx Input setup time before SCK rises	12	–	9.7	–	ns
$QSPI_4$	QIOx Input hold time after SCK rises	0.8	–	0.5	–	ns
$QSPI_5$	SCK rising to QIOx valid	0	2.7	0	2.1	ns
Mode 2						
$QSPI_6$	QIOx Input setup time before SCK rises	1.1	–	0.8	–	ns
$QSPI_7$	QIOx Input hold time after SCK rises	0.8	–	0.5	–	ns
$QSPI_8$	SCK rising to QIOx valid	0	3	0	2.3	ns
Mode 3						
$QSPI_9$	QIOx Input setup time before SCK falls	12.2	–	10	–	ns
$QSPI_{10}$	QIOx Input hold time after SCK falls	1	–	0.7	–	ns
$QSPI_{11}$	SCK falling to QIOx valid	0	3	0	2.4	ns

Table 62-70. QSPI1 IOSET2 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
Mode 0						
$QSPI_0$	QIOx Input setup time before SCK falls	1.9	–	1.9	–	ns
$QSPI_1$	QIOx Input hold time after SCK falls	0.8	–	0.7	–	ns
$QSPI_2$	SCK falling to QIOx valid	0	1.5	0	1.9	ns
Mode 1						
$QSPI_3$	QIOx Input setup time before SCK rises	12.6	–	10.4	–	ns
$QSPI_4$	QIOx Input hold time after SCK rises	0.2	–	0.1	–	ns
$QSPI_5$	SCK rising to QIOx valid	0	1.9	0	1.8	ns
Mode 2						
$QSPI_6$	QIOx Input setup time before SCK rises	2.2	–	2.1	–	ns
$QSPI_7$	QIOx Input hold time after SCK rises	0.2	–	0.1	–	ns
$QSPI_8$	SCK rising to QIOx valid	0	2.2	0	2	ns
Mode 3						
$QSPI_9$	QIOx Input setup time before SCK falls	11.9	–	10.4	–	ns
$QSPI_{10}$	QIOx Input hold time after SCK falls	0.8	–	0.7	–	ns
$QSPI_{11}$	SCK falling to QIOx valid	0	1.3	0	1.9	ns

Table 62-71. QSPI1 IOSET3 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
Mode 0						
$QSPI_0$	QIOx Input setup time before SCK falls	1.8	–	1.8	–	ns
$QSPI_1$	QIOx Input hold time after SCK falls	0.9	–	0.7	–	ns
$QSPI_2$	SCK falling to QIOx valid	0	1.3	0	1.8	ns
Mode 1						
$QSPI_3$	QIOx Input setup time before SCK rises	13.1	–	10.9	–	ns
$QSPI_4$	QIOx Input hold time after SCK rises	0.4	–	0.2	–	ns
$QSPI_5$	SCK rising to QIOx valid	0	1.7	0	1.7	ns
Mode 2						
$QSPI_6$	QIOx Input setup time before SCK rises	2.2	–	2.1	–	ns
$QSPI_7$	QIOx Input hold time after SCK rises	0.4	–	0.2	–	ns
$QSPI_8$	SCK rising to QIOx valid	0	2	0	1.8	ns
Mode 3						
$QSPI_9$	QIOx Input setup time before SCK falls	12.5	–	11	–	ns
$QSPI_{10}$	QIOx Input hold time after SCK falls	0.9	–	0.7	–	ns
$QSPI_{11}$	SCK falling to QIOx valid	0	1.1	0	1.7	ns

62.17 TWI Timings

Figure 62-29. Two-wire Serial Bus Timing

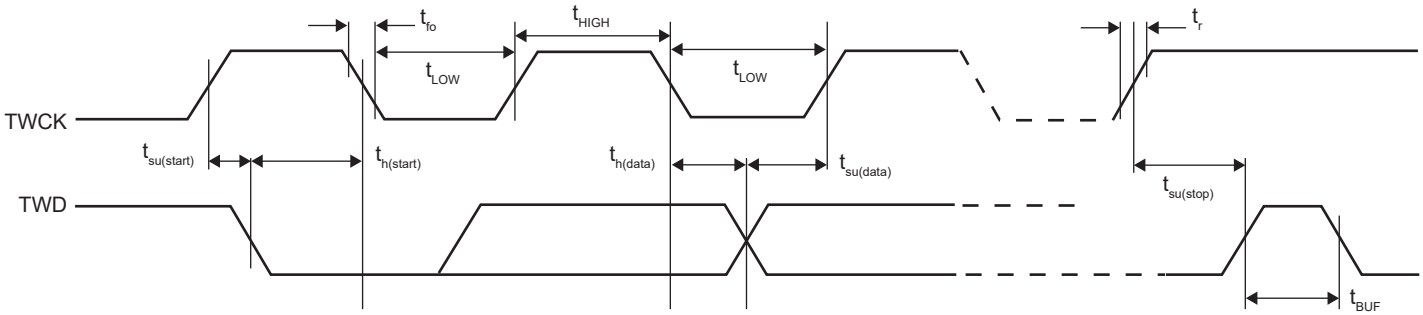


Table 62-72 describes the requirements for devices connected to the Two-wire Serial Bus.

Table 62-72. Two-wire Serial Bus Requirements

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IL}	Input Low-voltage	–	-0.3	$0.3 \times V_{DDIO}$	V
V_{IH}	Input High-voltage	–	$0.7 \times V_{DDIO}$	V_{CC} + 0.3	V
V_{hys}	Hysteresis of Schmitt Trigger Inputs	–	0.150	–	V
V_{OL}	Output Low-voltage	3 mA sink current	–	0.4	V
t_r	Rise Time for both TWD and TWCK		$20 + 0.1C_b$ (2)	300	ns
t_{fo}	Output Fall Time from V_{IHmin} to V_{ILmax}	$10\ pF < C_b < 400\ pF$ Figure 62-29	$20 + 0.1C_b$ (2)	250	ns
C_i (1)	Capacitance for each I/O Pin	–	–	10	pF
f_{TWCK}	TWCK Clock Frequency	–	0	400	kHz
Rp	Value of Pull-up Resistor	$f_{TWCK} \le 100$ kHz	$(V_{DDIO} - 0.4V) \div 3mA$	$1000ns \div C_b$	Ω
		$f_{TWCK} > 100$ kHz	$(V_{DDIO} - 0.4V) \div 3mA$	$300ns \div C_b$	Ω
t_{LOW}	Low Period of the TWCK Clock	$f_{TWCK} \le 100$ kHz	(3)	–	µs
		$f_{TWCK} > 100$ kHz	(3)	–	µs
t_{HIGH}	High Period of the TWCK Clock	$f_{TWCK} \le 100$ kHz	(4)	–	µs
		$f_{TWCK} > 100$ kHz	(4)	–	µs
$t_{h(start)}$	Hold Time (repeated) START condition	$f_{TWCK} \le 100$ kHz	t_{HIGH}	–	µs
		$f_{TWCK} > 100$ kHz	t_{HIGH}	–	µs
$t_{su(start)}$	Setup Time for a Repeated START condition	$f_{TWCK} \le 100$ kHz	t_{HIGH}	–	µs
		$f_{TWCK} > 100$ kHz	t_{HIGH}	–	µs
$t_{h(data)}$	Data Hold Time	$f_{TWCK} \le 100$ kHz	0	$(HOLD + 3) \times t_{peripheral\ clock}$	µs
		$f_{TWCK} > 100$ kHz	0	$(HOLD + 3) \times t_{peripheral\ clock}$	µs
$t_{su(data)}$	Data Setup Time	$f_{TWCK} \le 100$ kHz	$t_{LOW} - (HOLD + 3) \times t_{peripheral\ clock}$	–	ns
		$f_{TWCK} > 100$ kHz	$t_{LOW} - (HOLD + 3) \times t_{peripheral\ clock}$	–	ns

Table 62-72. Two-wire Serial Bus Requirements (Continued)

Symbol	Parameter	Conditions	Min	Max	Unit
$t_{su(stop)}$	Setup time for STOP condition	$f_{TWCK} \leq 100$ kHz	t_{HIGH}	–	µs
		$f_{TWCK} > 100$ kHz	t_{HIGH}	–	µs
t_{BUF}	Bus free time between a STOP and START condition	$f_{TWCK} \leq 100$ kHz	t_{LOW}	–	µs
		$f_{TWCK} > 100$ kHz	t_{LOW}	–	µs

Notes:
1. Required only for $f_{TWCK} > 100$ kHz
2. C_b = capacitance of one bus line in pF. Per I2C Standard, C_b Max = 400 pF
3. The TWCK low period is defined as follows: $t_{LOW} = ((CLDIV \times 2^{CKDIV}) + 4) \times t_{MCK}$
4. The TWCK high period is defined as follows: $t_{HIGH} = ((CHDIV \times 2^{CKDIV}) + 4) \times t_{MCK}$

62.18 MPDDRC Timings

62.18.1 Board Design Constraints

As the SAMA5D2 series embeds impedance calibrated pads, there are no capacitive constraints on DDR signals. However, a board must be designed and equipped in order to respect propagation time and intrinsic delay in the SDRAM device. In all cases, line length to memory device must not exceed 5 cm.

62.18.2 DDR2-SDRAM

Note: For DDR2 memory, the SHIFT_SAMPLING field value in the MPRDDRC_RD_DATA_PATH register must be configured to 1.

Table 62-73. System Clock Waveform Parameters

Symbol	Parameter	Conditions	Min	Max	Unit
t_{DDRCK}	DDRCK Cycle Time	VDDCORE[1.1V, 1.32V], T_A = 85°C	7.5	8.0	ns
		VDDCORE[1.2V, 1.32V], VDDIODDR[1.75V, 1.9V], T_A = 85°C	6.0	8.0	ns

62.18.3 LPDDR1-SDRAM

Note: For LPDDR1 memory, the SHIFT_SAMPLING field value in the MPRDDRC_RD_DATA_PATH register must be configured as follows:

SHIFT_SAMPLING = 0 for 0 < DDR_CLK < 94 MHz

SHIFT_SAMPLING = 1 for 94 MHz < DDR_CLK < 166 MHz

Table 62-74. System Clock Waveform Parameters

Symbol	Parameter	Conditions	Min	Max	Unit
t_{DDRCK}	DDRCK Cycle Time	VDDCORE[1.1V, 1.32V], T_A = 85°C	7.5	–	ns
t_{DDRCK}	DDRCK Cycle Time	VDDCORE[1.2V, 1.32V], VDDIODDR[1.75V, 1.9V], T_A = 85°C	6.4	–	ns

62.18.4 LPDDR2/LPDDR3-SDRAM

Note: For LPDDR2/LPDDR3 memory, the SHIFT_SAMPLING field value in the MPRDDRC_RD_DATA_PATH register must be configured as follows:

SHIFT_SAMPLING = 0 for 0 < DDR_CLK < 80 MHz

SHIFT_SAMPLING = 1 for 80 MHz < DDR_CLK < 166 MHz

Table 62-75. System Clock Waveform Parameters

Symbol	Parameter	Conditions	Min	Max	Unit
t_{DDRCK}	DDRCK Cycle Time	VDDCORE[1.1V, 1.32V], T_A = 85°C	7.5	–	ns
t_{DDRCK}	DDRCK Cycle Time	VDDCORE[1.1V, 1.32V], VDDIODDR[1.18V, 1.3V], T_A = 85°C	6.9	–	ns

62.18.5 DDR3/DDR3L-SDRAM

Note: For DDR3/DDR3L memory, the SHIFT_SAMPLING field value in the MPRDDRC_RD_DATA_PATH register must be configured to 2.

Table 62-76. System Clock Waveform Parameters

Symbol	Parameter	Conditions	Min	Max	Unit
t_{DDRCK}	DDRCK Cycle Time	VDDCORE[1.1V, 1.32V], T_A = 85°C	7.5	–	ns
t_{DDRCK}	DDRCK Cycle Time	VDDCORE[1.1V, 1.32V], VDDIODDR[1.18V, 1.3V], T_A = 85°C	6.9	–	ns

62.19 SSC Timings

62.19.1 Timing Conditions

Timings assuming a capacitance load are given in Table 62-77.

Table 62-77. Capacitance Load (SSC0 and SSC1)

Supply	Corner	
	Max	Min
3.3V	30 pF	5 pF
1.8V	20 pF	5 pF

62.19.2 Timing Extraction

Figure 62-30. SSC Transmitter, TK and TF in Output

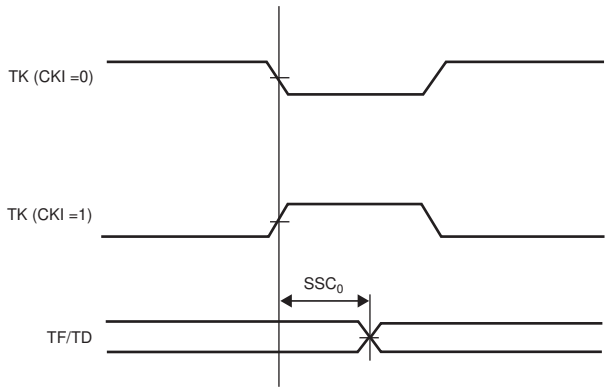


Figure 62-31. SSC Transmitter, TK in Input and TF in Output

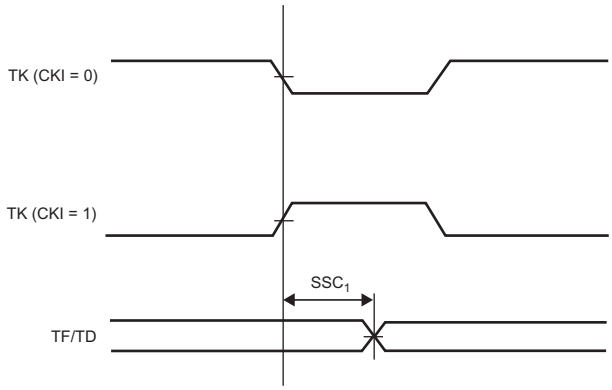


Figure 62-32. SSC Transmitter, TK in Output and TF in Input

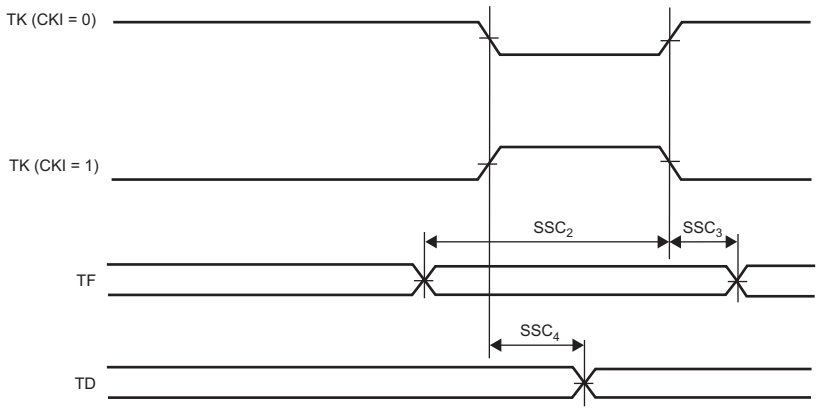


Figure 62-33. SSC Transmitter, TK and TF in Input

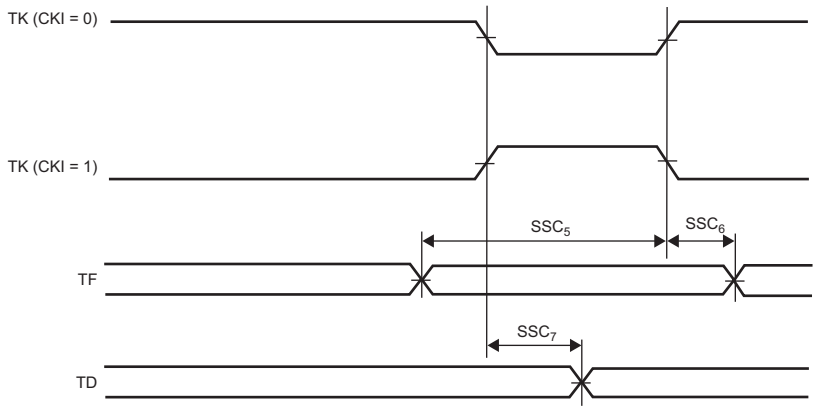


Figure 62-34. SSC Receiver RK and RF in Input

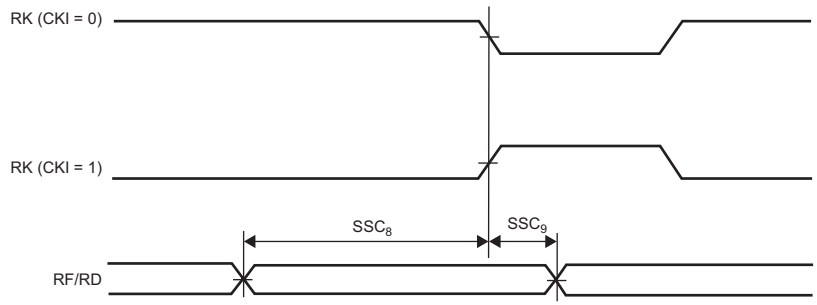


Figure 62-35. SSC Receiver, RK in Input and RF in Output

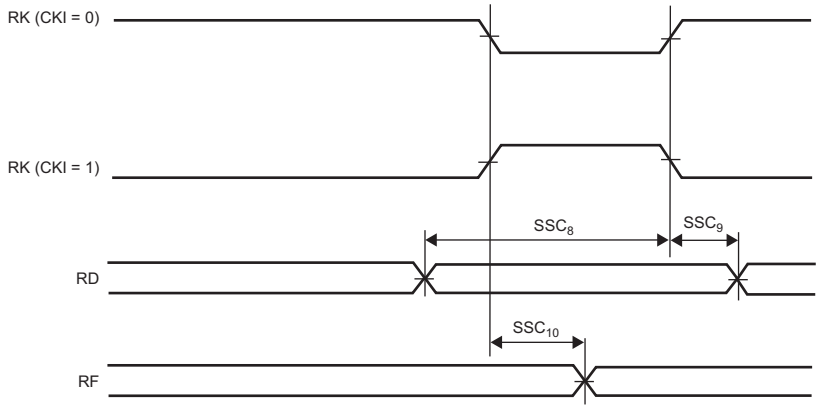


Figure 62-36. SSC Receiver, RK and RF in Output

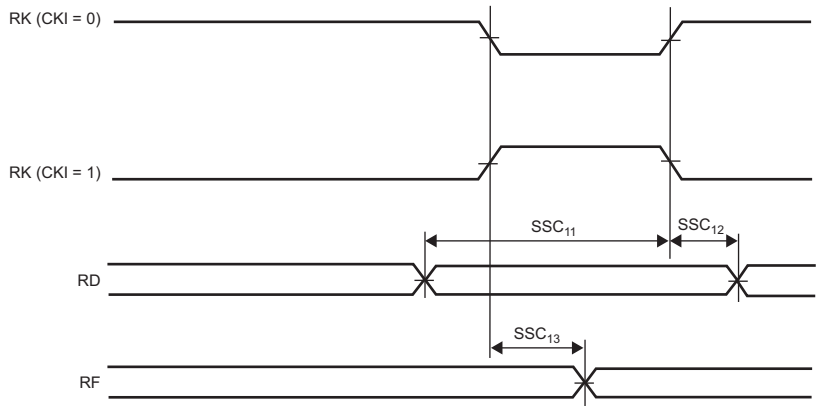


Figure 62-37. SSC Receiver, RK in Output and RF in Input

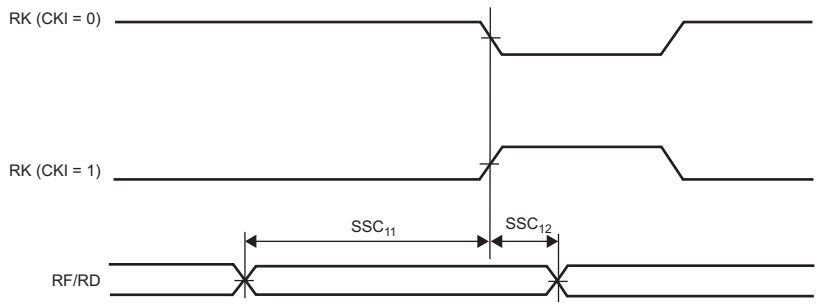


Table 62-78. SSC0 IOSET1 Timings

Symbol	Power supply / Parameter	Conditions	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Transmitter						
SSC_0	TK edge to TF/TD (TK output, TF output) (1)	–	0	3	0	3.3	ns
SSC_1	TK edge to TF/TD (TK input, TF output) (1)	–	3.7	13	3	11.3	ns
SSC_2	TF setup time before TK edge (TK output)	–	12.8	–	11.2	–	ns
SSC_3	TF hold time after TK edge (TK output)	–	0	–	0	–	ns
SSC_4	TK edge to TF/TD (TK output, TF input)(1)	–	0	3	0	3.3	ns
		STTDLY = 0 START = 4, 5 or 7	$2 \times t_{CPMCK}$	$3 + (2 \times t_{CPMCK})$	$2 \times t_{CPMCK}$	$3.3 + (2 \times t_{CPMCK})$	ns
SSC_5	TF setup time before TK edge (TK input)	–	0	–	0	–	
SSC_6	TF hold time after TK edge (TK input)	–	t_{CPMCK}	–	t_{CPMCK}	–	
SSC_7	TK edge to TF/TD (TK input, TF input)(1)	–	3.7	13	3.2	11.3	
		STTDLY = 0 START = 4, 5 or 7	$3.7 + (3 \times t_{CPMCK})$	$13 + (3 \times t_{CPMCK})$	$3.2 + (3 \times t_{CPMCK})$	$11.3 + (3 \times t_{CPMCK})$	
	Receiver						
SSC_8	RF/RD setup time before RK edge (RK input)	–	0	–	0	–	ns
SSC_9	RF/RD hold time after RK edge (RK input)	–	t_{CPMCK}	–	t_{CPMCK}	–	ns
SSC_{10}	RK edge to RF (RK input) (1)	–	3.5	12	2.9	10.4	ns
SSC_{11}	RF/RD setup time before RK edge (RK output)	–	$13.6 - t_{CPMCK}$	–	$12 - t_{CPMCK}$	–	ns
SSC_{12}	RF/RD hold time after RK edge (RK output)	–	t_{CPMCK}	–	t_{CPMCK}	–	ns
SSC_{13}	RK edge to RF (RK output) (1)	–	0	3	0	3.3	ns

Note: 1. For output signals (TF, TD, RF), minimum and maximum access times are defined. The minimum access time is the time between the TK (or RK) edge and the signal change. The maximum access time is the time between the TK edge and the signal stabilization. Figure 62-38 illustrates the minimum and maximum accesses for SSC_0. The same applies for SSC_1, SSC_4, SSC_7, SSC_{10} and SSC_{13}.

Table 62-79. SSC0 IOSET2 Timings

Symbol	Power supply / Parameter	Conditions	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
		Transmitter					
SSC_0	TK edge to TF/TD (TK output, TF output) (1)	–	0	3.4	0	3.7	ns
SSC_1	TK edge to TF/TD (TK input, TF output) (1)	–	3.5	12.3	2.8	10.5	ns
SSC_2	TF setup time before TK edge (TK output)	–	12	–	10.3	–	ns
SSC_3	TF hold time after TK edge (TK output)	–	0	–	0	–	ns
SSC_4	TK edge to TF/TD (TK output, TF input) (1)	–	0	3.4	0	3.5	ns
		STTDLY = 0 START = 4, 5 or 7	$2 \times t_{CPMCK}$	3.4 + $(2 \times t_{CPMCK})$	$2 \times t_{CPMCK}$	3.5 + $(2 \times t_{CPMCK})$	ns
SSC_5	TF setup time before TK edge (TK input)	–	0	–	0	–	
SSC_6	TF hold time after TK edge (TK input)	–	t_{CPMCK}	–	t_{CPMCK}	–	
SSC_7	TK edge to TF/TD (TK input, TF input) (1)	–	3.6	12.3	3	10.4	
		STTDLY = 0 START = 4, 5 or 7	3.6 + $(3 \times t_{CPMCK})$	12.3 + $(3 \times t_{CPMCK})$	3 + $(3 \times t_{CPMCK})$	10.4 + $(3 \times t_{CPMCK})$	
		Receiver					
SSC_8	RF/RD setup time before RK edge (RK input)	–	0	–	0	–	ns
SSC_9	RF/RD hold time after RK edge (RK input)	–	t_{CPMCK}	–	t_{CPMCK}	–	ns
SSC_{10}	RK edge to RF (RK input) (1)	–	3.3	11.5	2.7	9.8	ns
SSC_{11}	RF/RD setup time before RK edge (RK output)	–	13.8 - t_{CPMCK}	–	12.1 - t_{CPMCK}	–	ns
SSC_{12}	RF/RD hold time after RK edge (RK output)	–	t_{CPMCK}	–	t_{CPMCK}	–	ns
SSC_{13}	RK edge to RF (RK output) (1)	–	0	2.8	0	3.1	ns

Note: 1. For output signals (TF, TD, RF), minimum and maximum access times are defined. The minimum access time is the time between the TK (or RK) edge and the signal change. The maximum access time is the time between the TK edge and the signal stabilization. Figure 62-38 illustrates the minimum and maximum accesses for SSC_0. The same applies for SSC_1, SSC_4, SSC_7, SSC_{10} and SSC_{13}.

Table 62-80. SSC1 IOSET1 Timings

Symbol	Power supply / Parameter	Conditions	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Transmitter						
SSC_0	TK edge to TF/TD (TK output, TF output) (1)	–	0	2.6	0	2.7	ns
SSC_1	TK edge to TF/TD (TK input, TF output) (1)	–	3.6	12.7	3	10.9	ns
SSC_2	TF setup time before TK edge (TK output)	–	13.4	–	11.2	–	ns
SSC_3	TF hold time after TK edge (TK output)	–	0	–	0	–	ns
SSC_4	TK edge to TF/TD (TK output, TF input) (1)	–	0	2.1	0	2	ns
		STTDLY = 0 START = 4, 5 or 7	$2 \times t_{CPMCK}$	2.1 + $(2 \times t_{CPMCK})$	$2 \times t_{CPMCK}$	2 + $(2 \times t_{CPMCK})$	ns
SSC_5	TF setup time before TK edge (TK input)	–	0	–	0	–	
SSC_6	TF hold time after TK edge (TK input)	–	t_{CPMCK}	–	t_{CPMCK}	–	
SSC_7	TK edge to TF/TD (TK input, TF input) (1)	–	3.6	12.2	3	10.2	
		STTDLY = 0 START = 4, 5 or 7	3.6 + $(3 \times t_{CPMCK})$	12.2 + $(3 \times t_{CPMCK})$	3 + $(3 \times t_{CPMCK})$	10.2 + $(3 \times t_{CPMCK})$	
	Receiver						
SSC_8	RF/RD setup time before RK edge (RK input)	–	0	–	0	–	ns
SSC_9	RF/RD hold time after RK edge (RK input)	–	t_{CPMCK}	–	t_{CPMCK}	–	ns
SSC_{10}	RK edge to RF (RK input) (1)	–	3.4	11.8	2.7	9.9	ns
SSC_{11}	RF/RD setup time before RK edge (RK output)	–	12.2 - t_{CPMCK}	–	10.3 - t_{CPMCK}	–	ns
SSC_{12}	RF/RD hold time after RK edge (RK output)	–	t_{CPMCK}	–	t_{CPMCK}	–	ns
SSC_{13}	RK edge to RF (RK output) (1)	–	0	3.3	0	3.4	ns

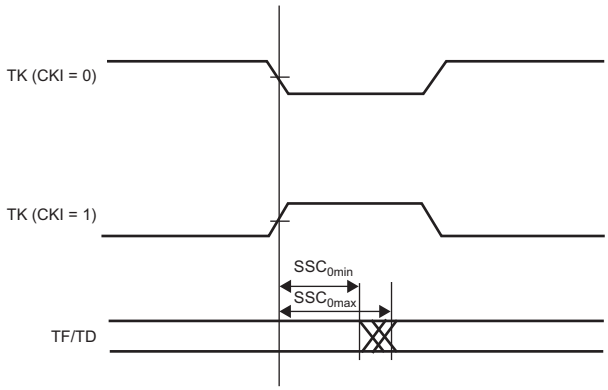
Note: 1. For output signals (TF, TD, RF), minimum and maximum access times are defined. The minimum access time is the time between the TK (or RK) edge and the signal change. The maximum access time is the time between the TK edge and the signal stabilization. Figure 62-38 illustrates the minimum and maximum accesses for SSC_0. The same applies for SSC_1, SSC_4, SSC_7, SSC_{10} and SSC_{13}.

Table 62-81. SSC1 IOSET2 Timings

Symbol	Power supply / Parameter	Conditions	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
			Transmitter				
SSC_0	TK edge to TF/TD (TK output, TF output) (1)	–	0	2.5	0	2.6	ns
SSC_1	TK edge to TF/TD (TK input, TF output) (1)	–	3.7	13	3.1	11.3	ns
SSC_2	TF setup time before TK edge (TK output)	–	14.3	–	12.2	–	ns
SSC_3	TF hold time after TK edge (TK output)	–	0	–	0	–	ns
SSC_4	TK edge to TF/TD (TK output, TF input) (1)	–	0	2.1	0	1.8	ns
		STTDLY = 0 START = 4, 5 or 7	$2 \times t_{CPMCK}$	2.1 + $(2 \times t_{CPMCK})$	$2 \times t_{CPMCK}$	1.8 + $(2 \times t_{CPMCK})$	ns
SSC_5	TF setup time before TK edge (TK input)	–	0	–	0	–	
SSC_6	TF hold time after TK edge (TK input)	–	t_{CPMCK}	–	t_{CPMCK}	–	
SSC_7	TK edge to TF/TD (TK input, TF input) (1)	–	3.7	12.6	3.1	10.4	
		STTDLY = 0 START = 4, 5 or 7	3.7 + $(3 \times t_{CPMCK})$	12.6 + $(3 \times t_{CPMCK})$	3.1 + $(3 \times t_{CPMCK})$	10.4 + $(3 \times t_{CPMCK})$	
			Receiver				
SSC_8	RF/RD setup time before RK edge (RK input)	–	0	–	0	–	ns
SSC_9	RF/RD hold time after RK edge (RK input)	–	t_{CPMCK}	–	t_{CPMCK}	–	ns
SSC_{10}	RK edge to RF (RK input) (1)	–	3.6	12.2	3	10.3	ns
SSC_{11}	RF/RD setup time before RK edge (RK output)	–	12.3 - t_{CPMCK}	–	10.5 - t_{CPMCK}	–	ns
SSC_{12}	RF/RD hold time after RK edge (RK output)	–	t_{CPMCK}	–	t_{CPMCK}	–	ns
SSC_{13}	RK edge to RF (RK output) (1)	–	0	2.9	0	3.1	ns

Note: 1. For output signals (TF, TD, RF), minimum and maximum access times are defined. The minimum access time is the time between the TK (or RK) edge and the signal change. The maximum access time is the time between the TK edge and the signal stabilization. Figure 62-38 illustrates the minimum and maximum accesses for SSC_0. The same applies for SSC_1, SSC_4, SSC_7, SSC_{10} and SSC_{13}.

Figure 62-38. Minimum and Maximum Access Time of Output Signals



62.20 PDMIC Timings

62.20.1 Timing Conditions

Timings assuming capacitance loads are given in Table 62-82.

Table 62-82. Capacitance Load

Supply	Corner	
	Max	Min
3.3V	30 pF	5 pF
1.8V	20 pF	5 pF

62.20.2 Timing Extraction

Figure 62-39. PDMIC Timing Diagram

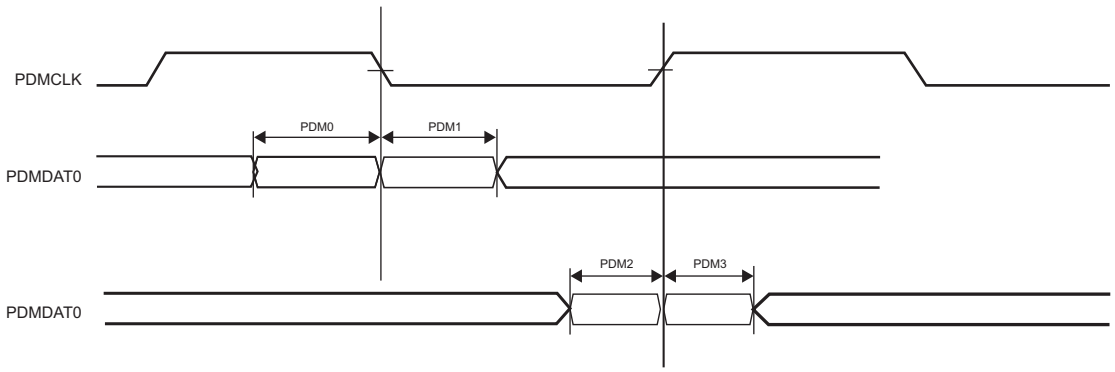


Table 62-83. PDMIC IOSET1 Timings

Symbol	Power Supply	1.8V		3.3V		Unit
	Parameter	Min	Max	Min	Max	
$PDMIC_0$	DATA setup time right	3.5	–	3.5	–	ns
$PDMIC_1$	DATA hold time right	3.1	–	3.5	–	ns
$PDMIC_2$	DATA setup time left	3.5	–	3.5	–	ns
$PDMIC_3$	DATA hold time left	3.1	–	3.5	–	ns

Table 62-84. PDMIC IOSET2 Timings

Symbol	Power Supply	1.8V		3.3V		Unit
	Parameter	Min	Max	Min	Max	
$PDMIC_0$	DATA setup time right	4.2	–	4.2	–	ns
$PDMIC_1$	DATA hold time right	2	–	2	–	ns
$PDMIC_2$	DATA setup time left	4.2	–	4.2	–	ns
$PDMIC_3$	DATA hold time left	2	–	2	–	ns

62.21 I2SC Timings

62.21.1 Timing Conditions

Timings assuming capacitance loads are given in Table 62-82.

Table 62-85. Capacitance Load (I2SC0 and I2SC1)

Supply	Corner	
	Max	Min
3.3V	30 pF	5 pF
1.8V	20 pF	5 pF

62.21.2 Timing Extraction

Table 62-86. I2SC0 IOSET1 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master					
$I2SC_0$	SDI Input setup time before SCK rises	12.5	–	10.8	–	ns
$I2SC_1$	SDI Input hold time after SCK rises	0	–	0	–	ns
$I2SC_2$	SCK falling to SDO valid	0	3.9	0	4	ns
$I2SC_3$	SCK falling to WS valid	0	2.7	0	3.1	ns
	Slave					
$I2SC_4$	SDI Input setup time before SCK rises	1.1	–	1	–	ns
$I2SC_5$	SDI Input hold time after SCK rises	1.3	–	1.2	–	ns
$I2SC_6$	WS Input setup time before SCK rises	2	–	1.8	–	ns
$I2SC_7$	WS Input hold time after SCK rises	0.9	–	0.8	–	ns
$I2SC_8$	SCK falling to SDO valid	4.2	14	3.6	12	ns

Table 62-87. I2SC0 IOSET2 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master					
$I2SC_0$	SDI Input setup time before SCK rises	11.7	–	9.7	–	ns
$I2SC_1$	SDI Input hold time after SCK rises	0	–	0	–	ns
$I2SC_2$	SCK falling to SDO valid	0	3	0	3	ns
$I2SC_3$	SCK falling to WS valid	0.1	4.7	0.2	4.8	ns
	Slave					
$I2SC_4$	SDI Input setup time before SCK rises	1.7	–	1.5	–	ns
$I2SC_5$	SDI Input hold time after SCK rises	0.6	–	0.4	–	ns
$I2SC_6$	WS Input setup time before SCK rises	3.6	–	3.4	–	ns
$I2SC_7$	WS Input hold time after SCK rises	0.7	–	0.6	–	ns
$I2SC_8$	SCK falling to SDO valid	3.7	12	3	10	ns

Table 62-88. I2SC1 IOSET1 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master					
$I2SC_0$	SDI Input setup time before SCK rises	13.1	–	11.4	–	ns
$I2SC_1$	SDI Input hold time after SCK rises	0	–	0	–	ns
$I2SC_2$	SCK falling to SDO valid	0	3.5	0	3.6	ns
$I2SC_3$	SCK falling to WS valid	0	2.9	0	3	ns
	Slave					
$I2SC_4$	SDI Input setup time before SCK rises	1.4	–	1.3	–	ns
$I2SC_5$	SDI Input hold time after SCK rises	1	–	0.8	–	ns
$I2SC_6$	WS Input setup time before SCK rises	2.4	–	2.1	–	ns
$I2SC_7$	WS Input hold time after SCK rises	0.8	–	0.7	–	ns
$I2SC_8$	SCK falling to SDO valid	4.4	13.8	3.7	11.9	ns

Table 62-89. I2SC1 IOSET2 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
	Master					
$I2SC_0$	SDI Input setup time before SCK rises	12.9	–	11.2	–	ns
$I2SC_1$	SDI Input hold time after SCK rises	0	–	0	–	ns
$I2SC_2$	SCK falling to SDO valid	0	3.6	0	3.7	ns
$I2SC_3$	SCK falling to WS valid	0	2.9	0	3	ns
	Slave					
$I2SC_4$	SDI Input setup time before SCK rises	1.1	–	1	–	ns
$I2SC_5$	SDI Input hold time after SCK rises	1.2	–	1	–	ns
$I2SC_6$	WS Input setup time before SCK rises	2.2	–	2	–	ns
$I2SC_7$	WS Input hold time after SCK rises	0.9	–	0.7	–	ns
$I2SC_8$	SCK falling to SDO valid	4.3	14	3.7	12	ns

62.22 ISC Timings

62.22.1 Timing Conditions

Timings assuming capacitance loads are given in Table 62-90.

Table 62-90. Capacitance Load

Supply	Corner	
	Max	Min
3.3V	30 pF	5 pF
1.8V	20 pF	5 pF

62.22.2 Timing Extraction

Figure 62-40. ISC Timing Diagram

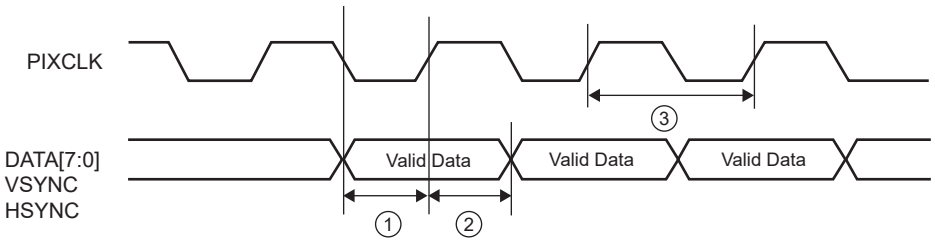


Table 62-91. ISC IOSET1 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
ISC_1	DATA/VSYNC/HSYNC setup time	3.9	–	3.6	–	ns
ISC_2	DATA/VSYNC/HSYNC hold time	0.7	–	0.6	–	ns
ISC_3	CONTROL VSYNC/HSYNC/FIELD setup time	4.4	–	4.2	–	ns
ISC_4	CONTROL VSYNC/HSYNC/FIELD hold time	0.4	–	0.3	–	ns
ISC_5	PIXCLK frequency	96	–	96	–	MHz

Table 62-92. ISC IOSET2 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
ISC_1	DATA/VSYNC/HSYNC setup time	4.3	–	4.2	–	ns
ISC_2	DATA/VSYNC/HSYNC hold time	0.5	–	0.3	–	ns
ISC_3	CONTROL VSYNC/HSYNC/FIELD setup time	4.6	–	4.4	–	ns
ISC_4	CONTROL VSYNC/HSYNC/FIELD hold time	0.2	–	0	–	ns
ISC_5	PIXCLK frequency	96	–	96	–	MHz

Table 62-93. ISC IOSET3 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
ISC_1	DATA/VSYNC/HSYNC setup time	4.6	–	4.2	–	ns
ISC_2	DATA/VSYNC/HSYNC hold time	0.5	–	0.4	–	ns
ISC_3	CONTROL VSYNC/HSYNC/FIELD setup time	4.3	–	4	–	ns
ISC_4	CONTROL VSYNC/HSYNC/FIELD hold time	1.5	–	0.4	–	ns
ISC_5	PIXCLK frequency	96	–	96	–	MHz

Table 62-94. ISC IOSET4 Timings

Symbol	Power Supply / Parameter	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
ISC_1	DATA/VSYNC/HSYNC setup time	4.3	–	4	–	ns
ISC_2	DATA/VSYNC/HSYNC hold time	0.5	–	0.4	–	ns
ISC_3	CONTROL VSYNC/HSYNC/FIELD setup time	4.2	–	4	–	ns
ISC_4	CONTROL VSYNC/HSYNC/FIELD hold time	0.5	–	0.3	–	ns
ISC_5	PIXCLK frequency	96	–	96	–	MHz

62.23 SDMMC Timings

The Secure Digital Multimedia Card (SDMMC) Controller supports the embedded MultiMedia Card (e.MMC) Specification V4.51, the SD Memory Card Specification V3.0, and the SDIO V3.0 specification. It is compliant with the SD Host Controller Standard V3.0 specification.

Features are different for the two instances of SDMMC:

SDMMC0: SD 3.0, eMMC 4.51, 8 bits

SDMMC1: SD 2.0, eMMC 4.41, 4 bits only

In SDR104 mode (SD 3.0), SDMMC0 is limited to 120 MHz (instead of 208 MHz). In HS200 mode (eMMC 4.51), SDMMC0 is limited to 120 MHz (instead of 200 MHz).

62.24 GMAC Timings

62.24.1 Timing Conditions

Timings assuming a capacitance load on data and clock are given in Table 62-95.

Table 62-95. Capacitance Load on Data, Clock Pads

Supply	Corner	
	Max	Min
3.3V	20 pF	0 pF

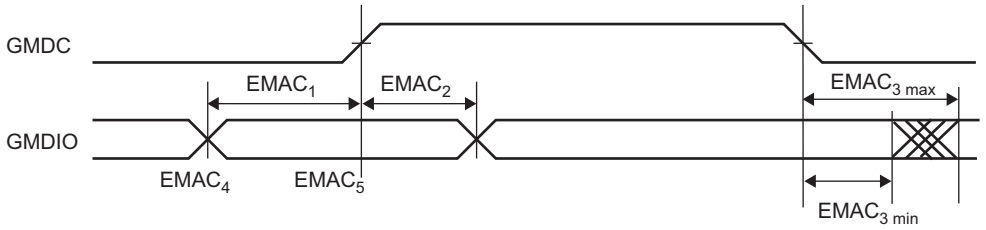
62.24.2 Timing Constraints

Table 62-96. Ethernet MAC Signals Relative to GMDC

Symbol	Parameter	Min	Max	Unit
$EMAC_1$	Setup for GMDIO from GMDC rising	10	–	ns
$EMAC_2$	Hold for GMDIO from GMDC rising	10	–	ns
$EMAC_3$	GMDIO toggling from GMDC rising(1)	0	300	ns

Note: 1. For Ethernet MAC output signals, minimum and maximum access time are defined. The minimum access time is the time between the GMDC rising edge and the signal change. The maximum access timing is the time between the GMDC rising edge and the signal stabilizes. Figure 62-41 illustrates minimum and maximum accesses for $EMAC_3$.

Figure 62-41. Minimum and Maximum Access Time of Ethernet MAC Output Signals

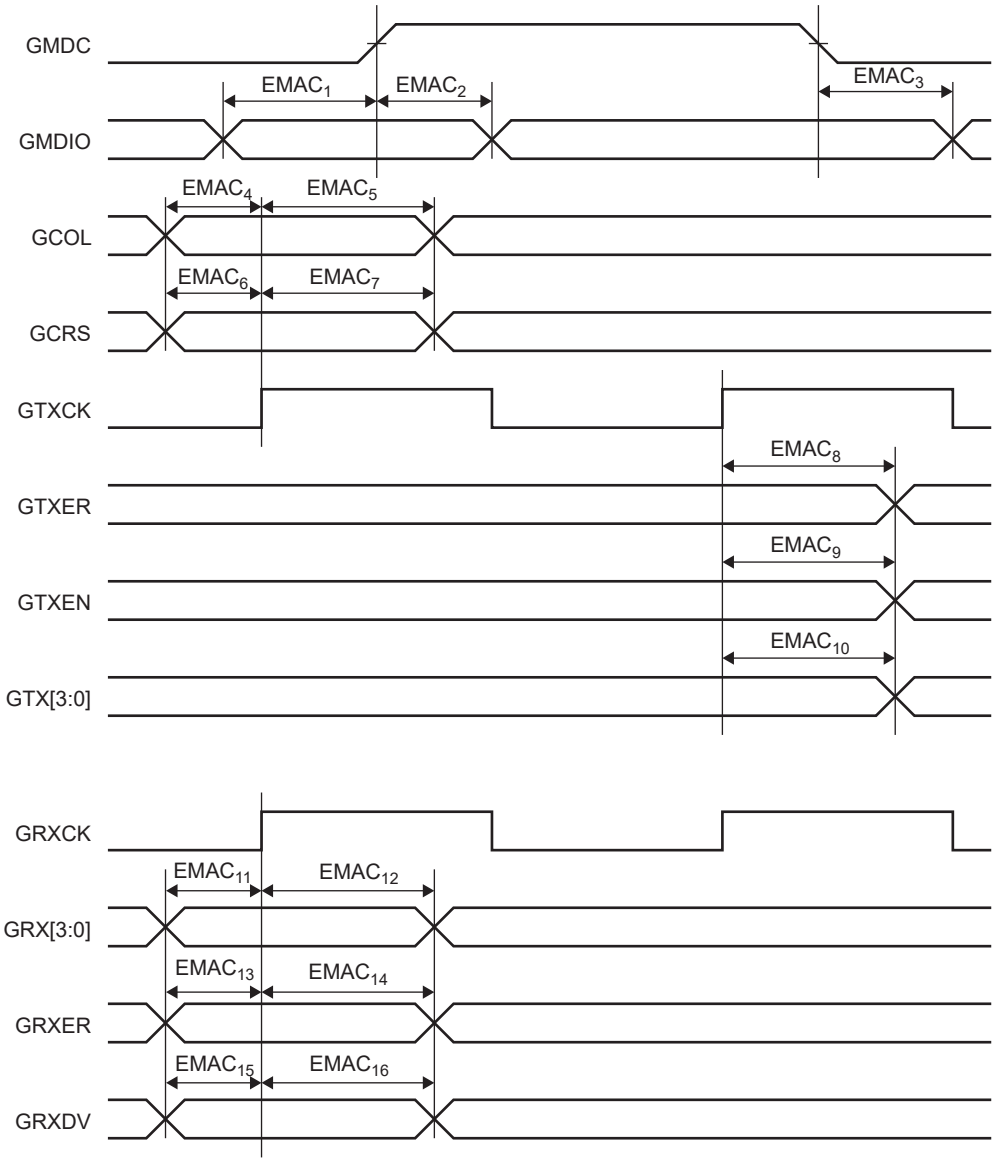


62.24.2.1 Ethernet MAC MII Mode

Table 62-97. Ethernet MAC MII Specific Signals

Symbol	Parameter	Min	Max	Unit
$EMAC_4$	Setup for GCOL from GTXCK rising	10	–	ns
$EMAC_5$	Hold for GCOL from GTXCK rising	10	–	ns
$EMAC_6$	Setup for GCRS from GTXCK rising	10	–	ns
$EMAC_7$	Hold for GCRS from GTXCK rising	10	–	ns
$EMAC_8$	GTXER toggling from GTXCK rising	10	25	ns
$EMAC_9$	GTXEN toggling from GTXCK rising	10	25	ns
$EMAC_{10}$	GTX toggling from GTXCK rising	10	25	ns
$EMAC_{11}$	Setup for GRX from GRXCK	10	–	ns
$EMAC_{12}$	Hold for GRX from GRXCK	10	–	ns
$EMAC_{13}$	Setup for GRXER from GRXCK	10	–	ns
$EMAC_{14}$	Hold for GRXER from GRXCK	10	–	ns
$EMAC_{15}$	Setup for GRXDV from GRXCK	10	–	ns
$EMAC_{16}$	Hold for GRXDV from GRXCK	10	–	ns

Figure 62-42. Ethernet MAC MII Mode

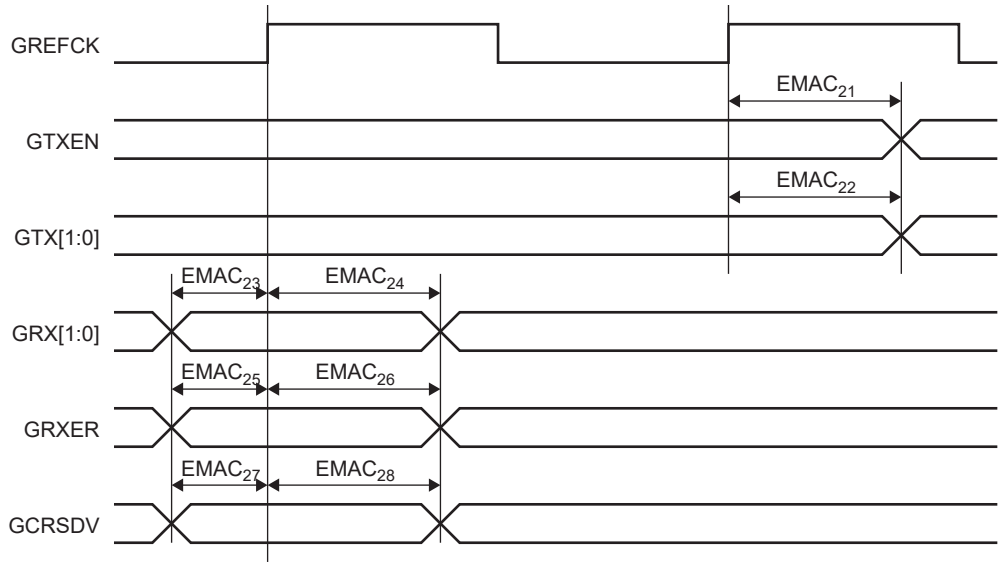


62.24.2.2 Ethernet MAC RMII Mode

Table 62-98. Ethernet MAC RMII Mode

Symbol	Parameter	Min	Max	Unit
$EMAC_{21}$	GTXEN toggling from GREFCK rising	2	16	ns
$EMAC_{22}$	GTX toggling from GREFCK rising	2	16	ns
$EMAC_{23}$	Setup for GRX from GREFCK rising	4	–	ns
$EMAC_{24}$	Hold for GRX from GREFCK rising	2	–	ns
$EMAC_{25}$	Setup for GRXER from GREFCK rising	4	–	ns
$EMAC_{26}$	Hold for GRXER from GREFCK rising	2	–	ns
$EMAC_{27}$	Setup for GCRSDV from GREFCK rising	4	–	ns
$EMAC_{28}$	Hold for GCRSDV from GREFCK rising	2	–	ns

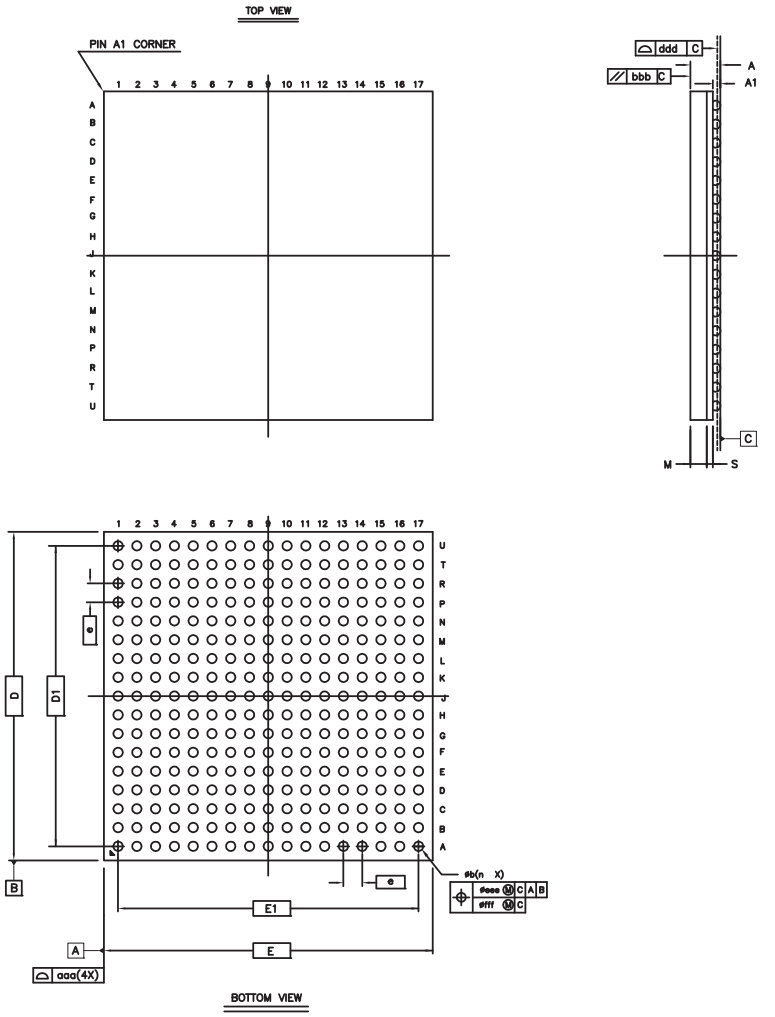
Figure 62-43. Ethernet MAC RMII Timings



63. Mechanical Characteristics

63.1 289-ball LFBGA Mechanical Characteristics

Figure 63-1. 289-ball LFBGA Package Drawing



		Symbol	Common Dimensions		
			MIN.	NOM.	MAX.
Package :			LFBGA		
Body Size:	X	E	14.000		
	Y	D	14.000		
Ball Pitch :		e	0.800		
Total Thickness :		A			1.400
Mold Thickness :		M	0.700 Ref.		
Substrate Thickness :		S	0.260 Ref.		
Ball Diameter :			0.400		
Stand Off :		A1	0.270	–	0.370
Ball Width :		b	0.380	–	0.480
Package Edge Tolerance :		aaa	0.150		
Mold Parallelism :		bbb	0.200		
Coplanarity:		ddd	0.120		
Ball Offset (Package) :		eee	0.150		
Ball Offset (Ball) :		fff	0.080		
Ball Count :		n	289		
Edge Ball Center to Center :	X	E1	12.800		
	Y	D1	12.800		

Table 63-1. 289-ball LFBGA Package Characteristics

Moisture Sensitivity Level	3

Table 63-2. Device and 289-ball LFBGA Package Weight

445	mg

Table 63-3. Package Reference

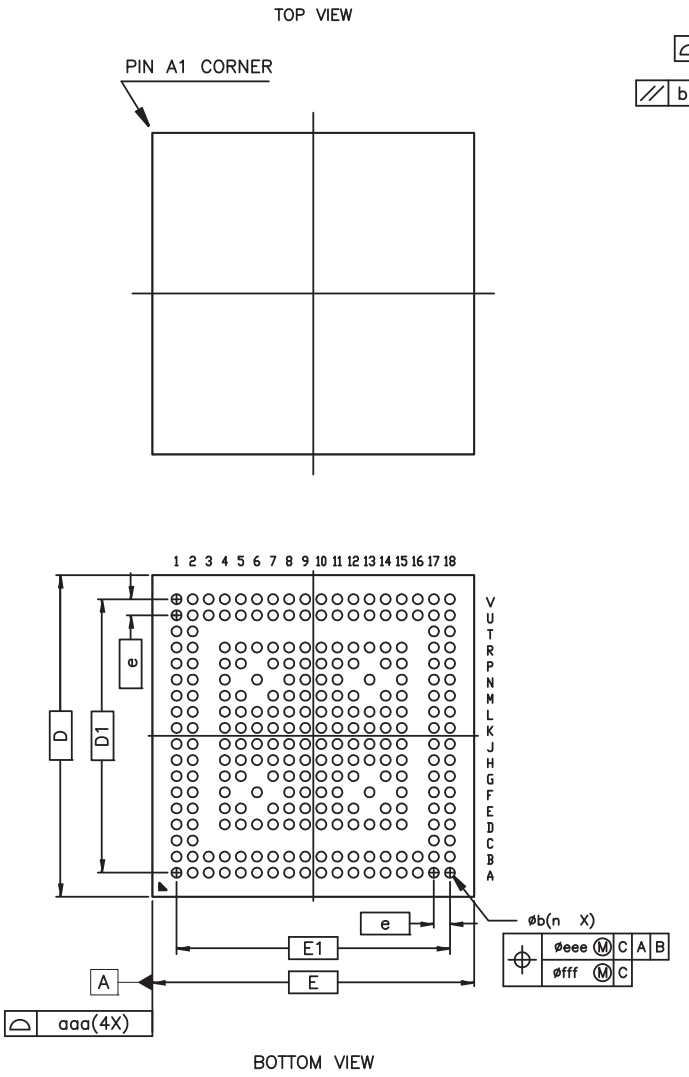
JEDEC Drawing Reference	NA
J-STD-609 Classification	e8

Table 63-4. 289-ball LFBGA Package Information

Ball Land	0.450 mm +/-0.05
Nominal Ball Diameter	0.4 mm
Solder Mask Opening	0.350 mm +/-0.05
Solder Mask Definition	SMD
Solder	OSP

63.2 256-ball TFBGA Mechanical Characteristics

Figure 63-2. 256-ball TFBGA Package



		Symbol	Common Dimensions		
			MIN.	NOM.	MAX.
Package :			TFBGA		
Body Size:	X	E		8.000	
	Y	D		8.000	
Ball Pitch :		e		0.400	
Total Thickness :		A			1.050
Mold Thickness :		M		0.530 Ref.	
Substrate Thickness :		S		0.260 Ref.	
Ball Diameter :				0.250	
Stand Off :		A1	0.110	–	0.210
Ball Width :		b	0.200	–	0.300
Package Edge Tolerance :		aaa		0.100	
Mold Parallelism :		bbb		0.100	
Coplanarity:		ddd		0.080	
Ball Offset (Package) :		eee		0.150	
Ball Offset (Ball) :		fff		0.080	
Ball Count :		n		256	
Edge Ball Center to Center :	X	E1		6.800	
	Y	D1		6.800	

Table 63-5. 256-ball TFBGA Package Characteristics

Moisture Sensitivity Level	3

Table 63-6. Device and 256-ball TFBGA Package Weight

110.3	mg

Table 63-7. Package Reference

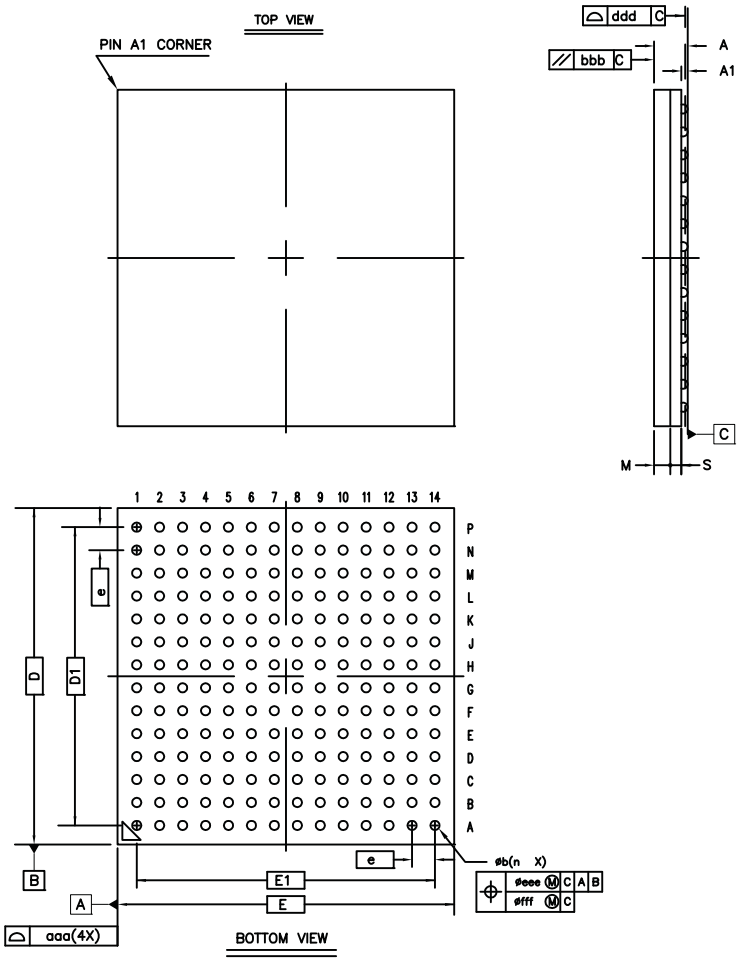
JEDEC Drawing Reference	NA
J-STD-609 Classification	e8

Table 63-8. 256-ball TFBGA Package Information

Ball Land	0.350 mm +/-0.05
Nominal Ball Diameter	0.25 mm
Solder Mask Opening	0.250 mm +/-0.05
Solder Mask Definition	SMD
Solder	OSP

63.3 196-ball TFBGA Mechanical Characteristics

Figure 63-3. Mechanical Overview of the 196-ball TFBGA Package



		Symbol	Common Dimensions		
			MIN.	NOM.	MAX.
Package :			TFBGA		
Body Size:	X	E	11.000		
	Y	D	11.000		
Ball Pitch :		e	0.750		
Total Thickness :		A			1.200
Mold Thickness :		M	0.530 Ref.		
Substrate Thickness :		S	0.360 Ref.		
Ball Diameter :			0.300		
Stand Off :		A1	0.160		0.260
Ball Width :		b	0.270		0.370
Package Edge Tolerance :		aaa	0.100		
Mold Parallelism :		bbb	0.100		
Coplanarity:		ddd	0.080		
Ball Offset (Package) :		eee	0.150		
Ball Offset (Ball) :		fff	0.080		
Ball Count :		n	196		
Edge Ball Center to Center :	X	E1	9.750		
	Y	D1	9.750		

Table 63-9. 196-ball TFBGA Package Characteristics

Moisture Sensitivity Level	3

Table 63-10. Device and 196-ball TFBGA Package Weight

234.2	mg

Table 63-11. Package Reference

JEDEC Drawing Reference	NA
J-STD-609 Classification	e8

Table 63-12. 196-ball TFBGA Package Information

Ball Land	0.350 mm +/-0.05
Nominal Ball Diameter	0.3 mm

Table 63-12. 196-ball TFBGA Package Information (Continued)

Solder Mask Opening	0.275 mm +/-0.30
Solder Mask Definition	SMD
Solder	OSP

64. Schematic Checklist

The schematic checklist provides the user with the requirements regarding the different pin connections that must be considered before starting any new board design as well as information on the minimum hardware resources required to quickly develop an application with the SAMA5D2. It does not consider PCB layout constraints.

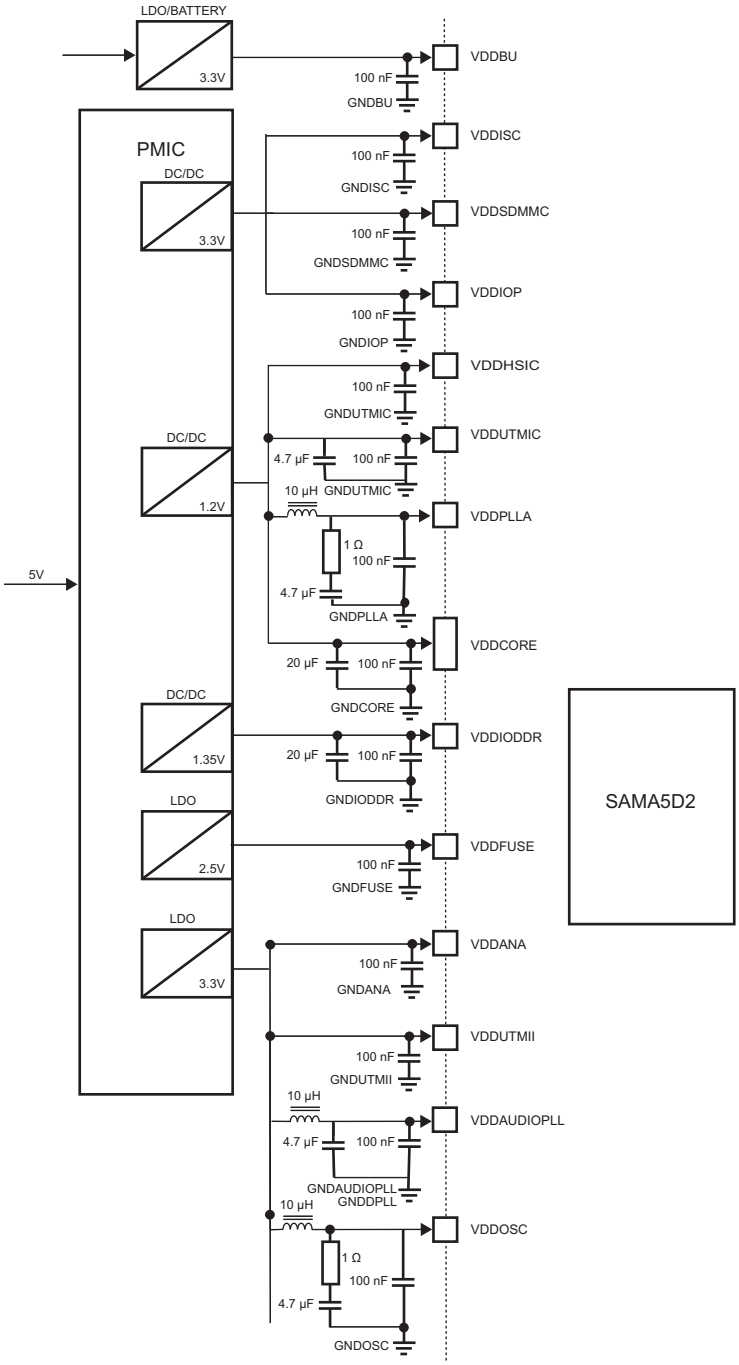
It also provides recommendations regarding low-power design constraints to minimize power consumption.

This information is not intended to be exhaustive. Its objective is to cover as many configurations of use as possible.

64.1 Power Supply

CAUTION: The board design must comply with the powerup and powerdown sequence guidelines provided in the datasheet to guarantee reliable operation of the device.

Figure 64-1. 1.2V, 1.35V/1.5V, 2V, 2.5V, 3.3V Power Supplies Schematics(1)



Note: 1. These values are given only as typical examples.

Table 64-1. Power Supply Connections

Signal Name	Recommended Pin Connection	Description
VDDCORE	1.08V to 1.32V Decoupling/Filtering capacitors (10 µF and 100 nF)[1][2]	Powers the core Supply ripple must not exceed 10 mVrms.
VDDPLLA	1.1V to 1.32V Decoupling capacitor (100 nF)[1][2]	Powers the PLLA cell. The VDDPLLA power supply pin draws small current, but it is noise-sensitive. Care must be taken in VDDPLLA power supply routing, decoupling and also on bypass capacitors. Supply ripple must not exceed 10 mVrms.
VDDIODDR	1.14V to 1.30V 1.283V to 1.45V 1.425V to 1.575V 1.7V to 1.95V Decoupling/Filtering capacitors (10 µF and 100 nF)[1][2]	Powers LPDDR2-LPDDR3 interface Powers DDR3L interface Powers DDR3 interface Powers LPDDR1-DDR2 interface Decoupling/filtering capacitors must be added to improve startup stability and reduce source voltage drop.
VDDISC	1.65V to 3.6V Decoupling capacitor (100 nF)[1][2]	Powers the ISC Interface I/O lines.
VDDIOP0,1,2	1.65V to 3.6V Decoupling capacitors (100 nF)[1][2]	Powers the peripherals I/O lines. Decoupling/filtering capacitors must be added to improve startup stability and reduce source voltage drop.
VDDBU	1.65V to 3.6V Decoupling capacitor (100 nF)[1][2]	Powers the Slow Clock oscillator, the internal 64 kHz RC and a part of the System Controller. Must be established first. Supply ripple must not exceed 30 mVrms.
VDDUTMIC	1.1V to 1.32V Decoupling capacitors (100 nF)[1][2]	DC Supply UTMI Phy (Core) and PLL UTMI Decoupling/filtering capacitors must be added to improve startup stability and reduce source voltage drop.
VDDUTMII	3.0V to 3.6V Decoupling capacitor (100 nF)[1][2]	DC Supply UTMI Phy (Interface) Decoupling/filtering capacitors must be added to improve startup stability and reduce source voltage drop.
VDDHSIC	1.1V to 1.3V Decoupling capacitors (100 nF)[1][2]	DC Supply HSIC Phy
VDDOSC	1.65V to 3.6V Decoupling/Filtering RLC circuit[1]	Powers the main oscillator cell. The VDDOSC power supply pin is noise-sensitive. Care must be taken in VDDOSC power supply routing, decoupling and also on bypass capacitors. Supply ripple must not exceed 30 mVrms.
VDDSDMMC	1.65V to 3.6V Decoupling capacitor (100 nF)[1][2]	Powers the SDMMC I/O lines.
VDDANA	1.65V to 3.6V Decoupling capacitor (100 nF)[1][2]	Powers the analog parts.
VDDFUSE	2.25V to 2.75V Decoupling capacitor (100 nF)[1][2]	Powers the fuse box for programming. VDDFUSE must not be left floating.

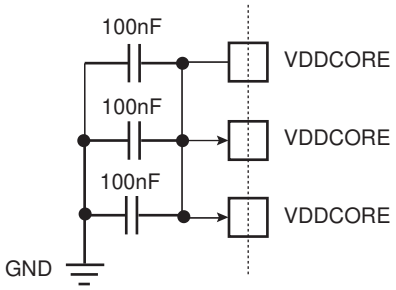
Table 64-1. Power Supply Connections (Continued)

Signal Name	Recommended Pin Connection	Description
VDDAUDIOPLL	3.0V to 3.6V Decoupling capacitor (100 nF)[1][2]	Powers the Audio PLL.
GNDCORE	Core Chip ground	GNDCORE pins are common to VDDCORE and VCCCORE pins. GNDCORE pins should be connected as shortly as possible to the system ground plane.
GNDPLLA	PLLA cell ground	GNDPLL pin is provided for VDDPLLA pins. GNDPLL pin should be connected as shortly as possible to the system ground plane.
GNDIODDR	DDR2/LPDDR1/LPDDR2/DDR3/LPDDR3 interface I/O lines ground	GNDIODDR pins should be connected as shortly as possible to the system ground plane.
GNDISC	VDDISC ground	GNDISC pins are common to VDDISC pins. GNDISC pins should be connected as shortly as possible to the system ground plane.
GNDIOP0,1,2	Peripherals and ISC I/O lines ground	GNDIOPx pins are common to VDDIOPx pins. GNDIOP pins should be connected as shortly as possible to the system ground plane.
GNDBU	Backup ground	GNDBU pin is provided for VDDBU pins. GNDBU pin should be connected as shortly as possible to the system ground plane.
GNDUTMIC	VDDUTMIC and VDDHSIC ground	GNDUTMIC pins are common to VDDUTMIC and VDDHSIC pins. GNDUTMIC pins should be connected as shortly as possible to the system ground plane.
GNDUTMII	UDPHS and UHPHS UTMI+ Core and Interface, and PLL UTMI ground	GNDUTMII pins are common to VDDUTMII and VDDUTMIC pins. GNDUTMII pins should be connected as shortly as possible to the system ground plane.
GNDOSC	Oscillator ground	GNDOSC pin is provided for VDDOSC pins. GNDOSC pin should be connected as shortly as possible to the system ground plane.
GNDSDMMC	SDMMC ground	SDMMC pins are common to VDDSDMMC pins. GNDSDMMC pins should be connected as shortly as possible to the system ground plane.
GNDANA	Analog ground	GNDANA pins are common to VDDANA pins. GNDANA pins should be connected as shortly as possible to the system ground plane.
GNDFUSE	Fuse box ground	GNDFUSE pins are common to VDDFUSE pins. GNDFUSE pins should be connected as shortly as possible to the system ground plane.

Table 64-1. Power Supply Connections (Continued)

Signal Name	Recommended Pin Connection	Description
GNDAUDIOPLL GNDDPLL	Audio PLL ground	GNDAUDIOPLL and GNDDPLL pins are common to VDDAUDIOPLL. GNDAUDIOPLL and GNDDPLL pins should be connected as shortly as possible to the system ground plane.

Note: 1. These values are given only as typical examples.
2. Decoupling capacitors must be connected as close as possible to the microprocessor and on each relevant pin.



For more information, see Table 62-37 “VDDCORE Power-On Reset Characteristics”.

64.2 Power-On Reset

The SAMA5D2 embeds several Power-On Resets (PORs) to ensure the power supply is established when the reset is released. These PORs are dedicated to VDDBU, VDDIOP and VDDCORE respectively.

64.3 Shutdown Considerations

When pin SHDN is asserted, NRST must be maintained to ‘L’ prior to remove the power supplies. After a delay of five SLCK periods, VDDPLL, then VDDCORE, then VDDIOP and VDDANA can be removed. At last, other power supplies can be removed.

VDDBU must never be removed when other supplies are present.

In case VDDBU is provided by a battery and the battery is discharged and reach the POR threshold (~1.4/1.6V), the SHDN pin MUST BE tied to 0 (reset state must be “0” level), in order to avoid restarting the external regulator if reset state is “1”.

64.4 Wakeup Considerations

When SHDN is rising, NRST is to be maintained to ‘L’ prior to establish the power supplies. Then VDDIOP and VDDANA are to be established, followed by VDDCORE and VDDPLL. At the end, other power supplies can be established. After a delay of five SLCK periods, the user can assert NRST to ‘H’ and make the system wake up. Asynchronous partial wakeup is also achievable using the RXLP or analog comparator (“RXLP” and “Analog comparator”).

64.5 Clock, Oscillator and PLL

Table 64-2. Clock, Oscillator and PLL Connections

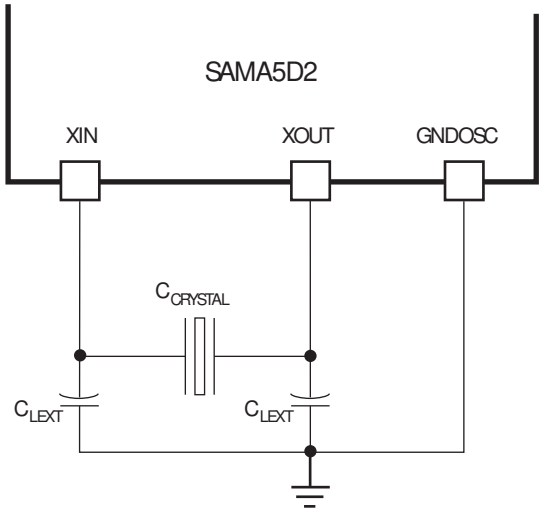
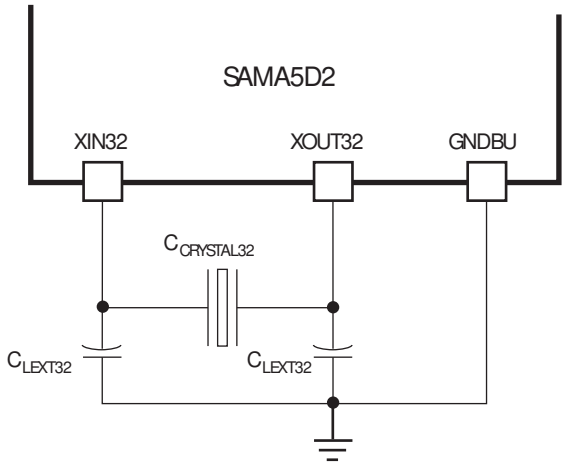
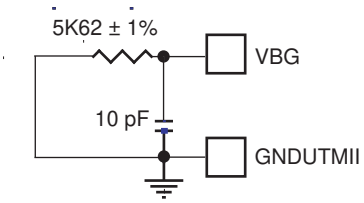
	Signal Name	Recommended Pin Connection	Description
	XIN XOUT 12 MHz Main Oscillator in Normal Mode	Crystals between 8 and 16 MHz USB High Speed (not Full Speed) Host and Device peripherals need a 12 MHz clock. Capacitors on XIN and XOUT (Crystal Load Capacitance dependent)	Crystal Load Capacitance to check ($C_{CRYSTAL}$)  Refer to Section 62. “Electrical Characteristics”.
	XIN XOUT 12 MHz Main Oscillator in Bypass Mode	XIN: external clock source XOUT: can be left unconnected USB High speed (not Full Speed) Host and Device peripherals need a 12 MHz clock.	Square wave signal, high level = VDDOSC External clock source up to 50 MHz Duty Cycle: 40 to 60% Refer to Section 62. “Electrical Characteristics”.
	XIN XOUT 12 MHz Main Oscillator Disabled	XIN: can be left unconnected XOUT: can be left unconnected USB High Speed (not Full Speed) Host and Device peripherals need a 12 MHz clock.	Typical nominal frequency 12 MHz (Internal 12 MHz RC Oscillator) Duty Cycle: 45 to 55% Refer to Section 62. “Electrical Characteristics”

Table 64-2. Clock, Oscillator and PLL Connections (Continued)

	Signal Name	Recommended Pin Connection	Description
	XIN32 XOUT32 Slow Clock Oscillator	32.768 kHz Crystal Capacitors on XIN32 and XOUT32 (Crystal Load Capacitance dependent)	Crystal load capacitance to check ($C_{CRYSTAL32}$)  Refer to Section 62. “Electrical Characteristics”.
	XIN32 XOUT32 Slow Clock Oscillator in Bypass Mode	XIN32: external clock source XOUT32: can be left unconnected	Square wave signal, high level = VDDBU External clock source up to 44 kHz Duty Cycle: 40 to 60% Refer to Section 62. “Electrical Characteristics”.
	XIN32 XOUT32 Slow Clock Oscillator Disabled	XIN32: can be left unconnected XOUT32: can be left unconnected	Typical nominal frequency 32 kHz (internal 64 kHz RC oscillator) Duty Cycle: 45 to 55% Refer to Section 62. “Electrical Characteristics”.
	VBG	0.9–1.1V(2)	Bias Voltage Reference for USB To reduce as much as possible the noise on VBG pin, check the layout considerations below: - VBG path as short as possible - Ground connection to GNDUTMII  VBG can be left unconnected if USB is not used. Refer to Section 4. “Signal Description”.

64.5.1 How to Define the Oscillator Load Capacitance

The load capacitance is the equivalent capacitor value that circuit must “show” to the crystal in order to oscillate at the target frequency. The load is set by two external capacitors placed on each side of the crystal to gnd.

The load of the crystal is the external capacitor value divided by two as it is represented in the figure below (in this case 2 * C_{LOAD} includes ALL the parasitics capacitors).

The C_{LOAD} value must be specified by the crystal manufacturer.

Parameters such as the parasitics of internal ASIC due to routing, IO pads, package and board must be calculated in order to reach the crystal load (refer to Section 62.7 “Oscillator Characteristics” to Section 62.7.3 “32.768 kHz Crystal Oscillator Characteristics”).

The external capacitor value can be determined by using the following formula:

$$C_{EXT} = (2 * C_{LOAD}) - C_{BOARD} - C_{PACKAGE} - C_{PAD} - C_{ROUTING} - (C_{PARA} * 2)$$

where:

- C_{EXT}: external capacitor value which must be soldered from aio33xin to gnd and from aio33xout to gnd
- C_{LOAD}: crystal-targeted load. Refer to C_{LOAD} parameter in the crystal electrical specification.
- C_{BOARD}: external calculated (or measured) value due to board parasitics
- $C_{PACKAGE}$: parasitics capacitance due to package and bonding
- C_{PAD}: parasitics capacitance of the I/O pad used for internal connection
- $C_{ROUTING}$: parasitics due to internal chip routing
- C_{PARA}: internal load parasitic due to internal structure. Refer to C_{PARA} parameter in Section 62.7 “Oscillator Characteristics”.

Table 64-3. Main Oscillator Load Capacitance

Oscillator	12 MHz < Frequency < 24 MHz	Frequency = 24 MHz
Main Oscillator	12.5 pF < C_{LOAD} < 17.5 pF	10 pF < C_{LOAD} < 12.5 pF

Table 64-4. 32.768 kHz Oscillator Load Capacitance

Oscillator	Frequency
32.768 kHz Oscillator	6 pF < C_{LOAD} < 12.5 pF

In our case, the minimum targeted load for the 32.768 kHz oscillator is 6 pF. The typical parasitic load of the oscillator is 0.5 pF. If routed + externals capacitances (package + board + external soldered + internal connection and internal pads parasitics) are about 1 pF, the external load must be 6 pF – 0.5 pF - 1 pF = 4.5 pF, which means that 9 pF is the target value (9 pF from aio33xin to gnd and 11 pF from aio33xout to gnd).

If a 12.5 pF load is targeted, the externals capacitances must be 12.5 pF - 0.5 pF - 1 pF = 11 pF, which is 22 pF (22 pF from aio33xin to gnd and 22 pF from aio33xout to gnd).

Example 1: Crystal C_{LOAD}= 6 pF
CI = 6 pF then Cxin_gnd_total = 12 pF and Cxout_gnd_total = 12 pF
9 pF external capacitor needed

Example 2: Crystal C_{LOAD} = 12.5 pF
CI = 12.5 pF then Cxin_gnd = 25 pF and Cxout_gnd = 25 pF
22 pF external capacitor needed

64.6 ICE and JTAG

Table 64-5. ICE and JTAG Connections(1)

	Signal Name	Recommended Pin Connection	Description
	TCK	Pull-up (100 kΩ)(2) If Debug mode is not required, this pin can be used as GPIO.	This pin is a Schmitt trigger input. Internal pull-up resistor to V_{DDIOP} (100 kΩ).
	TMS	Pull-up (100 kΩ)(2) If Debug mode is not required, this pin can be used as GPIO.	This pin is a Schmitt trigger input. Internal pull-up resistor to V_{DDIOP} (100 kΩ).
	TDI	Pull-up (100 kΩ)(2) If Debug mode is not required, this pin can be used as GPIO.	This pin is a Schmitt trigger input. Internal pull-up resistor to V_{DDIOP} (100 kΩ).
	TDO	Floating If Debug mode is not required, this pin can be used as GPIO.	Output driven at up to V_{DDIOP}.
	NTRST	Refer to the pin description section. If Debug mode is not required, this pin can be used as GPIO.	This pin is a Schmitt trigger input. Internal pull-up resistor to V_{DDIOP} (100 kΩ).
	JTAGSEL	In harsh environments(3), it is strongly recommended to tie this pin to GNDBU if not used or to add an external low-value resistor (such as 1 kΩ).	Internal pull-down resistor to GNDBU (15 kΩ). Must be tied to V_{DDBU} to enter JTAG Boundary Scan.

Notes:
1. It is recommended to establish accessibility to a JTAG connector for debug in any case.
2. These values are given only as typical examples.
3. In a well-shielded environment subject to low magnetic and electric field interference, the pin may be left unconnected. In noisy environments, a connection to ground is recommended.

64.7 Reset and Test

Table 64-6. Reset and Test Connections

	Signal Name	Recommended Pin Connection	Description
	NRST	Application-dependent Can be connected to a pushbutton for hardware reset.	NRST pin is a Schmitt trigger input. No internal pull-up resistor.
	TST	In harsh environments(1), it is strongly recommended to tie this pin to GNDBU to add an external low-value resistor (such as 10 kΩ).	This pin is a Schmitt trigger input. Internal pull-down resistor to GNDBU (15 kΩ).

Note: 1. In a well-shielded environment subject to low magnetic and electric field interference, the pin may be left unconnected. In noisy environments, a connection to ground is recommended.

64.8 Shutdown/Wakeup Logic

Table 64-7. Shutdown/Wakeup Logic Connections

	Signal Name	Recommended Pin Connection	Description
	SHDN	Application-dependent A typical application connects pin SHDN to the shutdown input of the DC/DC Converter providing the main power supplies.	This pin is a push-pull output. SHDN pin is driven low to GNDBU by the Shutdown Controller (SHDWC).
	WKUP	0 V to V_{DDBU}	This pin is an input only. WKUP behavior can be configured through the Shutdown Controller (SHDWC).

64.9 Parallel Input/Output (PIO)

Table 64-8. PIO Connections

	Signal Name	Recommended Pin Connection	Description
	PAx PBx PCx PDx	Application-dependent	All PIOs are pulled up inputs (100 kΩ) at reset except those which are multiplexed with the Address Bus signals that require to be enabled as peripherals: In Section 5. “Package and Pinout”, refer to the column ‘Reset State’ of the Pin Description table. Schmitt trigger on all inputs. To reduce power consumption if not used, the relevant PIO can be configured as an output, driven at ‘0’ with internal pull-up disabled.

64.10 Analog-to-Digital Converter (ADC)

Table 64-9. ADC Connections

	Signal Name	Recommended Pin Connection	Description
	ADVREF	3.3 V to VDDANA Decoupling/filtering capacitors Application-dependent	ADVREF is a pure analog input. To reduce power consumption if the ADC is not used, connect ADVREF to GNDANA.

64.11 External Bus Interface (EBI)

Table 64-10. EBI Connections

	Signal Name	Recommended Pin Connection	Description
	D0–D15	Application-dependent	Data Bus (D0 to D15) All data lines are pulled up inputs at reset.
	A0–A25	Application-dependent	Address Bus (A0 to A25) All address lines are pulled up inputs at reset.

Table 64-11 and Table 64-12 detail the connections to be applied between the EBI pins and the external devices for each memory controller.

Table 64-11. EBI Pins and External Static Devices Connections

Signals: EBI_	Pins of the Interfaced Device		
	8-bit Static Device	2 x 8-bit Static Devices	16-bit Static Device
Controller	**SMC (Static Memory Controller)**		
D0–D7	D0–D7	D0–D7	D0–D7
D8–D15	–	D8–D15	D8–D15
A0/NBS0	A0	–	NLB
A1	A1	A0	A0
A2–A22	A[2:22]	A[1:21]	A[1:21]
A23–A25	A[23:25]	A[22:24]	A[22:24]
NCS0	CS	CS	CS
NCS1	CS	CS	CS
NCS2	CS	CS	CS
NCS3/NANDCS	CS	CS	CS
NRD/NANDOE	OE	OE	OE
NWE/NWR0/NANDWE	WE	WE(1)	WE
NWR1/NBS1	–	WE(1)	NUB

Note: 1. NWR0 enables lower byte writes. NWR1 enables upper byte writes.

Table 64-12. EBI Pins and NAND Flash Device Connections

Signals: EBI_	Pins of the Interfaced Device	
	8-bit NAND Flash	16-bit NAND Flash
Controller	**NFC (NAND Flash Controller)**	
D0–D7	NFD0–NFD7	NFD0–NFD7
D8–D15	–	NFD8–NFD15
A21/NANDALE	ALE	ALE
A22/NANDCLE	CLE	CLE
NRD/NANDOE	RE	RE
NWE/NWR0/NANDWE	WE	WE

Table 64-12. EBI Pins and NAND Flash Device Connections (Continued)

Signals: EBI_	Pins of the Interfaced Device	
	8-bit NAND Flash	16-bit NAND Flash
Controller	**NFC (NAND Flash Controller)**	
NCS3/NANDCS	CE	CE
NANDRDY	R/$\overline{B}$	R/$\overline{B}$
A0/NBS0	–	–
A1–A20	–	–
A23–A25	–	–
NWR1/NBS1	–	–
NCS0	–	–
NCS1	–	–
NCS2	–	–
NWAIT	–	–

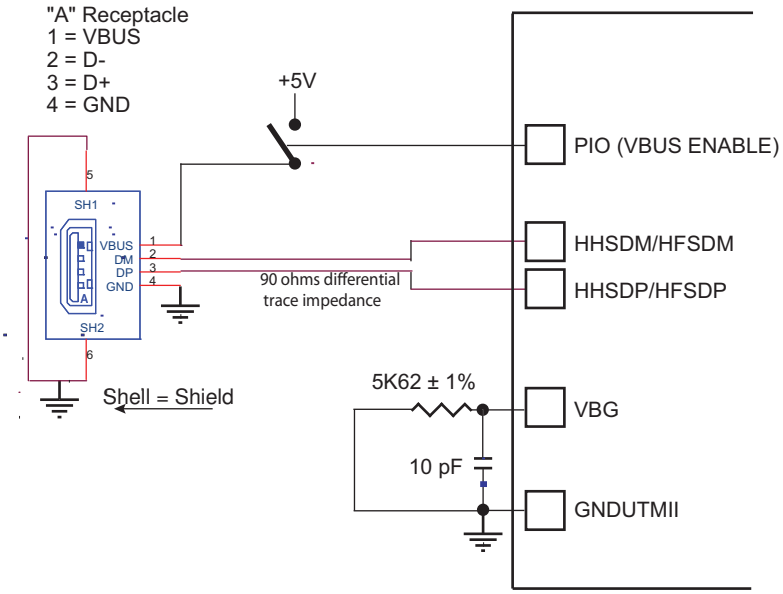
64.12 USB High-Speed Host Port (UHPHS) / USB High-Speed Device Port (UDPHS)

Table 64-13. UHPHS/UDPHS Connections

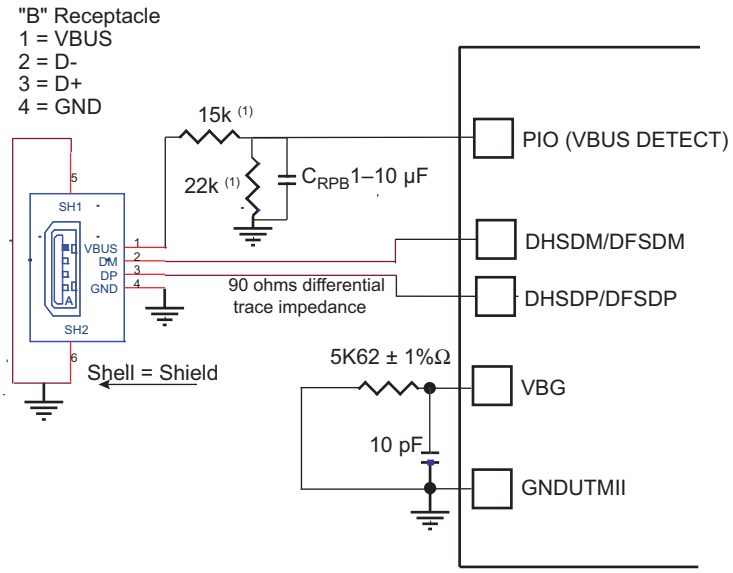
	Signal Name	Recommended Pin Connection	Description
UHPHS	HHSDPA/DHSDPB[1] HHSDMA/DHSDMB[1]	Application-dependent[2][3]	Pull-down output at reset
UDPHS	HHSDP/HHSDM	Application-dependent[2]	Pull-down output at reset
HSIC	HHSTROBE/HHDATA	Application-dependent[2]	Pull-down output at reset

Notes: 1. UDPHS shares Port A with UHPHS.

2. Example of a USB High Speed Host connection: refer to section Section 39. “USB Host High Speed Port (UHPHS)”.



3. Typical USB High Speed Device connection: refer to Section 38. “USB High Speed Device Port (UDPHS)”.



Note: 1. The values shown on the 22 kΩ and 15 kΩ resistors are only valid with 3.3-V supplied PIOs.

64.13 Boot Program Hardware Constraints

Refer to Section 15. “Standard Boot Strategies” for more details on the boot program.

64.14 Layout and Design Constraints

Note: All PIOs shared, multiplexed, connected to various components and connectors must be connected through serial resistors 22R and 39R for clock signals. The resistors must be populated as close as possible to the SAMA5D2.

64.14.1 General Considerations

This chapter provides routing guidelines for layout and design of a printed circuit board using high-speed interfaces, Serial, Ethernet and USB 2.0. The signal integrity rules for high-speed interfaces need to be considered. In fact, it is highly recommended that the board design be simulated to determine optimum layout for signal integrity and quality. Keep in mind that this document can only highlight the most important issues that should be considered when designing the SAMA5D2 board. The designer has to take into account the corresponding information (specification, design guidelines, etc.) contained in the documentation of all other devices that are to be implemented on board.

64.14.2 Considerations for High-Speed Differential Interfaces

The following is a list of suggestions for designing with high-speed differential signals.

This should help implementing these interfaces while providing maximum SAMA5D2 board performance.

- Use controlled impedance PCB traces that match the specified differential impedance.
- Keep the trace lengths of the differential signal pairs as short as possible.
- The differential signal pair traces should be trace-length matched and the maximum trace-length mismatch should not exceed the specified values. Match each differential pair per segment.
- Maintain parallelism and symmetry between differential signals with the trace spacing needed to achieve the specified differential impedance.
- Maintain maximum possible separation between the differential pairs and any high-speed clocks/periodic signals (CMOS/TTL) and any connector leaving the PCB (such as I/O connectors, control and signal headers, or power connectors).
- Route differential signals on the signal layer nearest to the ground plane using a minimum of vias and corners. This will reduce signal reflections and impedance changes. Use GND stitching vias when changing layers.
- It is best to put CMOS/TTL and differential signals on different layers which should be isolated by the power and ground planes.
- Avoid tight bends. When it becomes necessary to turn 90°, use two 45° turns or an arc instead of making a single 90° turn.
- Do not route traces under crystals, crystal oscillators, clock synthetizers, magnetic devices or ICs that use, and/or generate, clocks.
- Stubs on differential signals should be avoided due to the fact that stubs cause signal reflections and affect signal quality.
- Keep the length of high-speed clock and periodic signal traces that run parallel to high-speed signal lines at a minimum to avoid crosstalk. Based on EMI testing experience, the minimum suggested spacing to clock signals is 50 mil.
- Use a minimum of 20 mil spacing between the differential signal pairs and other signal traces for optimal signal quality. This helps to prevent crosstalk.
- Route all traces over continuous planes (VCC or GND) with no interruptions.
- Avoid crossing over anti-etch if at all possible. Crossing over anti-etch (split planes) increases inductance and radiation levels by forcing a greater loop area.

64.14.3 DDR Layout and Design Considerations

Refer to document “SAMA5D2 Layout Recommendations”, Atmel literature No. 44041.

64.14.4 e.MMC routing

Refer to the Micron Technical Note TN-FC-35: e•MMC PCB Design Guide. This document is intended as guide for PCB designers using Micron e•MMC devices and discusses the primary issues affecting design and layout.

64.14.5 USB Trace Routing Guidelines

- Maintain parallelism between USB differential signals with the trace spacing needed to achieve 90-ohm differential impedance. Deviations will normally occur due to package breakout and routing to connector pins. Just ensure the amount and lengths of the deviations are kept to the minimum possible.
- Use an impedance calculator to determine the trace width and spacing required for the specific board stack-up being used. Example: Layer Stacking, 7.5-mil traces with 7.5-mil spacing results in approximately 90-ohm differential trace impedance.
- Minimize the length of high-speed clock and periodic signal traces that run parallel to high-speed USB signal lines, to minimize crosstalk. Based on EMI testing experience, the minimum suggested spacing to clock signals is 50 mils.
- Based on simulation data, use 20-mil minimum spacing between high-speed USB signal pairs and other signal traces for optimal signal quality. This helps to prevent crosstalk.

Table 64-14. USB Trace Routing Guidelines

Parameter	Trace Routing
Signal length allowance for the SAMA5D2	14.0 inches Valid for a damping value of the PCB trace of 0.11 dB/inch @ 0.4 GHz (common value for FR-4 based material)
Differential impedance	90 ohms +/-15%
Single-ended impedance	45 ohms +/-10%
Trace width (W)	5 mils (microstrip routing)
Spacing between differential pairs (intra-pair)	6 mils (microstrip routing)
Spacing between pairs (inter-pair)	Min. 20 mils
Spacing between differential pairs and high-speed periodic signals	Min. 50 mils
Spacing between differential pairs and low-speed non-periodic signals	Min. 20 mils
Length matching between differential pairs (intra-pair)	150 mils
Reference plane	GND referenced preferred
Spacing from edge of plane	Min. 40 mils
Vias usage	Try to minimize the number of vias

64.14.6 QSPI Pull-up Resistors

The ROM code **removes** the internal pull-up resistors when it configures PIO controller to mux the QSPI controller I/O lines. Therefore the probing step may fail if the Quad I/O mode of the memory has not been enabled yet and if this memory does not embed internal pull-up resistor on #HOLD or #RESET pin.

This is why we recommend to add external pull-up resistors if needed on the four I/O data lines MOSI/IO0, MISO/IO1, #WP/IO2 and #HOLD/IO3.

Another solution is to update the Quad Enable non-volatile bit in the relevant register to reassign #WP and #HOLD/#RESET pins to functions IO2 and IO3.

65. Marking

All devices are marked with the Atmel logo and the ordering code.

Additional marking is as follows:

where

- “YY”: Manufactory year
- “WW”: Manufactory week
- “C”: Assembly country code
- “V”: Revision
- “XXXXXX”: Lot number

66. Ordering Information

Table 66-1. SAMA5D2 Ordering Information

Ordering Code	MRL	Package	Carrier Type	Operating Temperature Range
ATSAMA5D22A-CU	A	TFBGA196	Tray	-40°C to 85°C
ATSAMA5D22A-CUR	A		Tape and reel	
ATSAMA5D24A-CU	A	TFBGA256	Tray	
ATSAMA5D24A-CUR	A		Tape and reel	
ATSAMA5D27A-CU	A	LFBGA289	Tray	
ATSAMA5D27A-CUR	A		Tape and reel	
ATSAMA5D28A-CU[1]	A		Tray	
ATSAMA5D21B-CU	B	TFBGA196	Tray	-40°C to 85°C
ATSAMA5D21B-CUR	B		Tape and reel	
ATSAMA5D22B-CN	B		Tray	-40°C to 105°C
ATSAMA5D22B-CNR	B		Tape and reel	
ATSAMA5D22B-CU	B		Tray	-40°C to 85°C
ATSAMA5D22B-CUR	B		Tape and reel	
ATSAMA5D23B-CN	B		Tray	-40°C to 105°C
ATSAMA5D23B-CNR	B		Tape and reel	
ATSAMA5D23B-CU	B		Tray	-40°C to 85°C
ATSAMA5D23B-CUR	B		Tape and reel	
ATSAMA5D24B-CU	B	TFBGA256	Tray	-40°C to 85°C
ATSAMA5D24B-CUR	B		Tape and reel	
ATSAMA5D26B-CN	B	LFBGA289	Tray	-40°C to 105°C
ATSAMA5D26B-CNR	B		Tape and reel	
ATSAMA5D26B-CU	B		Tray	-40°C to 85°C
ATSAMA5D26B-CUR	B		Tape and reel	
ATSAMA5D27B-CN	B		Tray	-40°C to 105°C
ATSAMA5D27B-CNR	B		Tape and reel	
ATSAMA5D27B-CU	B		Tray	-40°C to 85°C
ATSAMA5D27B-CUR	B		Tape and reel	
ATSAMA5D28B-CN	B		Tray	-40°C to 105°C
ATSAMA5D28B-CNR	B		Tape and reel	
ATSAMA5D28B-CU	B		Tray	-40°C to 85°C
ATSAMA5D28B-CUR	B		Tape and reel	

Note: 1. To order ATSAMA5D28 and ATSAMA5D23, contact an Atmel Sales Representative.

67. Errata

Errata is described in the following sections:

67.1 Errata - SAMA5D2 MRL-B Parts

This section describes errata relevant to the devices listed in Table 67-1.

Table 67-1. SAMA5D2 MRL-B Parts

Device Name
ATSAMA5D21B
ATSAMA5D22B
ATSAMA5D23B
ATSAMA5D24B
ATSAMA5D26B
ATSAMA5D27B
ATSAMA5D28B

67.1.1 ROM Code: SDMMC0 and SDMMC1 boot issue

Issue: **The card detect pin is not correctly sampled in the ROM code, which leads to a nondeterministic boot ability on the SDMMC0/SDMMC1 interfaces (SDCard or eMMC).**

Workaround: Use another boot media (e.g., serial flash) for the first level boot, and either deactivate the boot on SDMMC0/1 in the Boot Configuration Word in the Fuse area or remove any bootable program stored in the eMMC or SDCard connected to the chip at startup.

67.1.2 SDMMC Software ‘Reset for All’ Command

Issue: **Software ‘Reset for All’ command is not guaranteed**

The software ‘Reset for All’ command is not guaranteed, and some registers of the host controller may not properly reset. The setting of the different registers must be checked before reinitializing the SD card.

Workaround: None

67.1.3 FLEXCOM SMBUS

Issue: **FLEXCOM SMBUS alert signalling is not functional**

The TWI function embedded in the FLEXCOM does not support SMBUS alert signal management.

Workaround: If this signal is mandatory in the application, the user can use one of the standalone TWIs (TWIHS0, TWIHS1) supporting the SMBUS alert signaling.

67.1.4 TWI/TWIHS Clear Command

Issue: **The TWI/TWIHS Clear command does not work**

Bus reset using the “CLEAR” bit of the TWI/TWIHS control register does not work correctly during a bus busy state..

Workaround: When the TWI master detects the SDA line stuck in low state the procedure to recover is:

1. Reconfigure the SDA/SCL lines as PIO.
2. Try to assert a Logic 1 on the SDA line (PIO output = 1).
3. Read the SDA line state. If the PIO state is a Logic 0, then generate a clock pulse on SCL (1-0-1 transition).
4. Read the SDA line state. If the SDA line = 0, go to Step 3; if SDA = 1, go to Step 5.
5. Generate a STOP condition.
6. Reconfigure SDA/SCL PIOs as peripheral.

67.1.5 SSC TD Output

Issue: **Unexpected delay on TD output**

When SSC is configured with the following conditions:

- RCMR.START = Start on falling edge/Start on Rising edge/Start on any edge,
- RFMR.FSOS = None (input),
- TCMR.START = Receive Start,

an unexpected delay of 2 or 3 system clock cycles is added to the TD output.

Workaround: None

67.1.6 I2SC First Sent Data

Issue: **I2SC first sent data corrupted**

Right after I2SC reset, the first data sent by I2SC controller on the I2SDO line is corrupted. The following data are not affected.

Workaround: None

67.1.7 Quad I/O Serial Peripheral Interface (QSPI)

Issue: **QSPI hangs with long DLYCS**

QSPI hangs if a command is written to any QSPI register during the DLYCS delay. There is no status bit to flag the end of the delay.

Workaround: The field DLYCS defines a minimum period for which Chip Select is deasserted, required by some memories. This delay is generally < 60 ns and comprises internal execution time, arbitration and latencies. Thus, DLYCS must be configured to be slightly higher than the value specified for the slave device. The software must wait for this same period of time plus an additional delay before a command can be written to the QSPI.

67.1.8 Master/Processor Clock Prescaler

Issue: Change of the field PMC_MCKR.PRES is not allowed if Master/Processor Clock Prescaler frequency is too high

PMC_MCKR.PRES cannot be changed if the clock applied to the Master/Processor Clock Prescaler (see “Master Clock Controller”, in Section 30. “Power Management Controller (PMC)”) is greater than 312 MHz (VDDCORE[1.1, 1.32]) and 394 MHz (VDDCORE[1.2, 1.32]).

Workaround:

1. Set PMC_MCKR.CSS to MAIN_CLK.
2. Set PMC_MCKR.PRES to the required value.
3. Change PMC_MCKR.CSS to the new clock source (PLLA_CLK, UPLLCK).

67.2 Errata - SAMA5D2 MRL-A Parts

This section describes errata relevant to the devices listed in Table 67-2.

Table 67-2. SAMA5D2 MRL-A Parts

Device Name
ATSAMA5D22A
ATSAMA5D24A
ATSAMA5D27A
ATSAMA5D28A

67.2.1 ROM Code: Main External Clock Frequency Support for SAM-BA Monitor

Issue: **ROM code v1.1 supports ONLY a 12 and 16 MHz external clock frequency to allow USB connection to be used for SAM-BA Monitor**

Workaround: None

67.2.2 ROM Code: Watchdog after SAM-BA Monitor Connection

Issue: **Watchdog reset occurs when reenabling the watchdog**

When no bootable program is found in an external memory, the Watchdog is disabled just before the ROM Code runs SAM-BA Monitor. The ROM code sets the Watchdog Timer Mode register (WDT_MR) to the value 0x00008000 and then clears the counter value. If a program loaded and executed using the SAM-BA Monitor Go command reenables the watchdog, a watchdog reset is immediately executed whatever the value of the watchdog counter.

Workaround: To avoid any unexpected watchdog reset when reenabling the watchdog, the following sequence has to be performed:

4. Write 0x00000000 in the WDT_MR.
5. Wait for three slow clock cycles.
6. Write the final value in the WDT_MR.

67.2.3 Unique Serial Number

Issue: **The serial number stored in the SFR registers (SFR_SN0 and SFR_SN1) is not correct**

The serial number (SFR_SN0, SFR_SN1) has only 16 bits set. This serial number cannot be used as a 64-bit unique ID.

Workaround: None

67.2.4 Fuse Masking

Issue: **The Partial Fuse Masking function does not work**

The fuse masking function described in Section 54. “Secure Fuse Controller (SFC)” does not work. If the ROM code is used in Secure mode, the overall fuses are masked by the ROM code even if some of them are not used.

Workaround: None

67.2.5 Fuse Writing

Issue: **The first two bits of each 32-bit block of the fuse matrix cannot be written**

The first two bits of each 32-bit block of the fuse matrix cannot be written, so that any word (32 bits) written needs to set to 0 the first two bits of each word (32 bits) of the fuse matrix.

Workaround: None

67.2.6 HSIC Startup

Issue: **At HSIC startup, the strobe default state is wrong**

The strobe line should be at logic state 0 when HSIC is powered ON, and disabled. Currently, powering up the product sets the strobe line at logic state 1 before the HSIC is enabled. In this case, a connected device tries to connect before the HSIC is enabled.

Workaround: Connect the device after the SAMA5D2 has been started.

67.2.7 ADC SleepWalking

Issue: **ADC SleepWalking is not functional**

Workaround: None

67.2.8 ADC Last Channel Low-speed Trigger

Issue: **The last channel can be triggered at low speed but cannot be programmed by the OUT1 field of the RTC. Only the 1-Hz sampling period is available.**

Workaround: None

67.2.9 ADC Trigger Events

Issue: **ADC trigger events RTCOUT0 and RTCOUT1 are not functional**

RTCOUT0 issue leads to ADC Sleepwalking not functional.

RTCOUT1 issue makes the last channel specific measurement trigger work at 1 Hz only.

Workaround: None

67.2.10 ACC Output

Issue: **ACC output connection issue**

The Analog Comparator (ACC) output is not connected to the PWM event line.

Workaround: None

67.2.11 SDMMC Software 'Reset For all' Command

Issue: **Software 'Reset For all' command is not guaranteed**

The software 'Reset For all' command is not guaranteed, and some registers of the host controller may not properly reset. The setting of the different registers must be checked before reinitializing the SD card.

Workaround: None

67.2.12 SDMMC Status Flag INTCLKS

Issue: **Status flag INTCLKS may not work correctly**

When the SDMMC internal clock is disabled (SDMMC_CCR. INTCLKEN = 0) and reenabled after a few cycles (SDMMC_CCR. INTCLKEN = 1), the status flag INTCLKS may get stuck at 0.

Workaround: A delay loop of 6 cycles minimum of the slowest clock (HCLOCK or BASECLK) must be inserted between SDMMC_CCR. INTCLKEN = 0 and SDMMC_CCR. INTCLKEN = 1.

67.2.13 PMC GCLK Fields

Issue: **GCLK fields are reprogrammed unexpectedly**

When configuring a peripheral featuring no GCLK, the GCLK fields (GCKEN, GCKCSS, GCKDIV) of FLEXCOM0 are reconfigured. No other parameter is modified.

Workaround: When accessing a peripheral featuring no GCLK, fill the GCLK fields with the GCLK configuration of FLEXCOM0.

67.2.14 PMC SleepWalking

Issue: **PMC SleepWalking is not functional**

In Ultra-Low Power mode (ULP1) using simultaneously partial wakeup (SleepWalking) and full wakeup (PIOBU used as wakeup pins or internal events RTC, etc.) may not resume from ULP1.

Workaround: None.

67.2.15 CLASSD Peripheral

Issue: **Unexpected offset and noise level in Differential Output mode**

When the CLASSD peripheral is set to Differential Output mode (PWMTYP = 1), a significant output offset and an increased level of noise are observed on the audio outputs. The offset is systematic and is equal to 1/16 of the digital full scale.

Workaround: To avoid the offset, add the opposite offset on the input signal of the CLASSD peripheral.

67.2.16 FLEXCOM SMBUS

Issue: **FLEXCOM SMBUS alert signalling is not functional**

The TWI function embedded in the FLEXCOM does not support SMBUS alert signal management.

Workaround: If this signal is mandatory in the application, the user can use one of the standalone TWIs (TWIHS0, TWIHS1) supporting the SMBUS alert signaling.

67.2.17 TWI/TWIHS Clear Command

Issue: **The TWI/TWIHS Clear command does not work**

Bus reset using the “CLEAR” bit of the TWI/TWIHS control register does not work correctly during a bus busy state..

Workaround: When the TWI master detects the SDA line stuck in low state the procedure to recover is:

1. Reconfigure the SDA/SCL lines as PIO.
2. Try to assert a Logic 1 on the SDA line (PIO output = 1).
3. Read the SDA line state. If the PIO state is a Logic 0, then generate a clock pulse on SCL (1-0-1 transition).
4. Read the SDA line state. If the SDA line = 0, go to Step 3; if SDA = 1, go to Step 5.
5. Generate a STOP condition.
6. Reconfigure SDA/SCL PIOs as peripheral.

67.2.18 MPDDRC t_{FAW}

Issue: **t_{FAW} timing violation**

DDR2/LPDDR2 memory devices with 8 banks have an additional requirement for t_{FAW}: no more than four Activate commands must be issued in any given t_{FAW} period.

Workaround: Increase the value of t_{RRD} to 3 to avoid the issue.

67.2.19 Audio PLL

Issue: **Audio PLL output frequency range**

The frequency range of the AUDIOCORECLK signal (AUDIOPLL output) provided in Table 62-23 “Audio PLL Characteristics” (f_{CORE} parameter) does not comply with the applicable specification.

Workaround: The AUDIOCORECLK signal can be operated from 720 MHz to 790 MHz if the following restricted operating conditions are met:

- Junction temperature (T_j) range: 0°C to +40°C
- VDDCORE/VDDPLL supply range: 1.20V to 1.32V
- Bits <29:28> in register PMC_AUDIO_PLL0 are set to $(01)_2$

67.2.20 SSC TD Output

Issue: **Unexpected delay on TD output**

When SSC is configured with the following conditions:

- RCMR.START = Start on falling edge/Start on Rising edge/Start on any edge,
- RFMR.FSOS = None (input),
- TCMR.START = Receive Start,

an unexpected delay of 2 or 3 system clock cycles is added to the TD output.

Workaround: None

67.2.21 I2SC First Sent Data

Issue: **I2SC first sent data corrupted**

Right after I2SC reset, the first data sent by I2SC controller on the I2SDO line is corrupted. The following data are not affected.

Workaround: None

67.2.22 Quad I/O Serial Peripheral Interface (QSPI)

Issue: QSPI hangs with long DLYCS

QSPI hangs if a command is written to any QSPI register during the DLYCS delay. There is no status bit to flag the end of the delay.

Workaround: The field DLYCS defines a minimum period for which Chip Select is de-asserted, required by some memories. This delay is generally < 60 ns and comprises internal execution time, arbitration and latencies. Thus, DLYCS must be configured to be slightly higher than the value specified for the slave device. The software must wait for this same period of time plus an additional delay before a command can be written to the QSPI.

67.2.23 Master/Processor Clock Prescaler

Issue: Change of the field PMC_MCKR.PRES is not allowed if Master/Processor Clock Prescaler frequency is too high

PMC_MCKR.PRES cannot be changed if the clock applied to the Master/Processor Clock Prescaler (see “Master Clock Controller”, in Section 30. “Power Management Controller (PMC)”) is greater than 312 MHz (VDDCORE[1.1, 1.32]) and 394 MHz (VDDCORE[1.2, 1.32]).

Workaround:

1. Set PMC_MCKR.CSS to MAIN_CLK.
2. Set PMC_MCKR.PRES to the required value.
3. Change PMC_MCKR.CSS to the new clock source (PLLA_CLK, UPLLCK).

68. Revision History

In the table that follows, the most recent version of the document appears first.

Table 68-1. SAMA5D2 Datasheet Rev. 11267E Revision History

Issue Date	Changes
25-Jul-16	Deleted Section 61. “Security Module”.

Table 68-2. SAMA5D2 Datasheet Rev. 11267D Revision History

Issue Date	Changes
12-May-16	Minor formatting and editorial changes throughout
	“Introduction” Updated listed DDR memories
	“Features” Frequency of digital fractional PLL for audio “11.289 MHz” corrected to “11.2896 MHz” “Two 64-bit, 16-channel DMA controllers” changed to “51 DMA Channels including two 16-channel 64-bit Central DMA Controllers”
	Section 1. “Description” Updated description of Low-power modes
	Section 2. “Configuration Summary” “Class D amplifier” changed to “stereo Class D amplifier” Updated text at end of section
	Section 3. “Block Diagram” Figure 3-1 “SAMA5D2 Series Block Diagram”: added ISC_MSK input; updated description of crystal oscillators; “PWMEXTRIG0-1” renumbered to “PWMEXTRG1–2” Added note “See Section 35. “DMA Controller (XDMAC)” for peripheral connections to DMA.”
	Section 4. “Signal Description” Table 4-1 “Signal Description List”: NRST signal function “Microcontroller Reset” changed to “Microprocessor Reset”; “PWMEXTRG0-1” renumbered to “PWMEXTRG1–2”; “Self-refresh mode” changed to “Backup Self-refresh mode” in DDR_CKE comments
	Section 5. “Package and Pinout” Separated content into Section 5.1 “Packages” and Section 5.2 “Pinouts” Table 5-2 “Pin Description (SAMA5D21, SAMA5D22, SAMA5D24, SAMA5D26, SAMA5D27, SAMA5D28A)”: “ADVREFP” corrected to “ADVREF”; “PWMEXTRG0” and “PWMEXTRG1” renumbered to “PWMEXTRG1” and “PWMEXTRG2”; removed empty function cells for primary signals PA30, PA31, and PB0–PB7; removed “SEC, FILTER” from “Reset State” column header; added footnote on reset states Added Table 5-3 “Pin Description (SAMA5D23 pins different from those in SAMA5D21/SAMA5D22)” and Table 5-4 “Pin Description (SAMA5D28B pins different from those in SAMA5D28A)”
	Section 6. “Power Considerations” Table 6-1 “SAMA5D2 Power Supplies”: updated rows VDDUTMIC, VDDHSIC and VDDOSC Section 6.4.1 “VDDBU Power Architecture”: reworded second paragraph and deleted “typically less than 2 µA”
	Section 7. “Memories” Section 7.2.1 “External Bus Interface”: “The slew rates are determined by programming the SFR_EBICFG bit in SFR registers” changed to “The drive levels are configured with the DRIVEx field in the EBI Configuration Register (SFR_EBICFG)”

Table 68-2. SAMA5D2 Datasheet Rev. 11267D Revision History (Continued)

Issue Date	Changes
12-May-16	Section 8. “Event System” Section 8-1 “Real-time Event Mapping List”: instance of “ADC_ADTRG” corrected to “ADTRG”
	Section 9. “System Controller” Section 9.1 “Power-On Reset”: “dedicated to VDDBU, VDDIOP and VDDCORE” changed to “dedicated to monitoring VDDBU, VDDIOP and VDDCORE”
	Section 10. “Peripherals” Table 10-1 “Peripheral identifiers”: in ‘Instance Name’ column, renamed CAN0 and CAN1 to MCAN0 and MCAN1 Section 10.4 “Peripheral Clock Types”: in SLOW_CLOCK description, “32768-Hz crystal oscillator or by the on-chip 32-kHz RC oscillator” changed to “32.768 kHz crystal oscillator or by the on-chip 64 kHz RC oscillator”
	Section 11. “Chip Identifier (CHIPID)” Updated Table 11-1 “SAMA5D2 Chip ID Registers”
	Section 13. “L2 Cache Controller (L2CC)” Table 13-2 “Register Mapping”: reset value 0x0000_0000 changed to 0x0000_0111 for L2CC_TRCR and L2CC_DRCR
	Section 14. “Debug and Test Features” Table 14-1 “Debug and Test Pin List”: NRST pin function “Microcontroller Reset” changed to “Microprocessor Reset”
	Section 15. “Standard Boot Strategies” “Boot Sequence Control Register (BSCR)” renamed to “Boot Sequence Controller Configuration Register” Section 15.1 “Description”: “This microcontroller can be configured” changed to “This microprocessor can be configured” Figure 15-10 “Galois Field Table Mapping”: modified Galois field table offsets Section 15.4.2 “Boot Sequence Controller Configuration Register”: added address Section 15.4.3 “Boot Configuration Word”: added reference to “Customer Fuse Matrix” Added Section 15.4.6.5 “QSPI Flash Boot” Table 15-3 “PIO Driven during Boot Program Execution”: NAND Flash PIO line PIOC17 changed to PIOB0
	Section 18. “Special Function Registers (SFR)” Table 18-1 “Register Mapping”: removed EBI Configuration Register / SFR_EBICFG (offset 0x40 now reserved) Section 18.3.1 “DDR Configuration Register”: added note Removed section “EBI Configuration Register”
	Section 21. “Watchdog Timer (WDT)” Section 21.4 “Functional Description”: in eighth paragraph, “To prevent a software deadlock that continuously triggers the watchdog, the reload of the watchdog must occur...” changed to “The reload of the watchdog must occur...”
	Section 25. “Real-time Clock (RTC)” Reworked Section 25.5.6 “Updating Time/Calendar” Reworked Figure 25-7 “Calibration Circuitry Waveforms” AD index ‘7’ replaced with generic ‘n’ in Section 25.5.8 “Waveform Generation” Updated Figure 25-8 “Waveform Generation for ADC Trigger Event” Section 25.6.2 “RTC Mode Register”: - updated descriptions of fields OUT0 and OUT1 - added fields TPERIOD and THIGH
	Section 27. “Low Power Asynchronous Receiver (RXLP)” Pin/signal name “LPRXD” changed to “RXD”

Table 68-2. SAMA5D2 Datasheet Rev. 11267D Revision History (Continued)

Issue Date	Changes
12-May-16	Section 29. “Clock Generator” Section 29.2 “Embedded Characteristics”: AUDIOPLLCK changed to AUDIOPLLCLK Figure 29-1 “Clock Generator Block Diagram”: lines changed to arrows for OSCSEL to multiplexer, for MOSCSEL to multiplexer, and for PLLADIV2 to “PLLA and Divider” block Figure 29-5 “Divider and PLLA Block Diagram”: added PLLADIV2 divider Updated Section 29.8 “Audio PLL”
	Section 30. “Power Management Controller (PMC)” AUDIOPLLCK changed to AUDIOPLLCLK in Section 30.15 “Programmable Clock Controller” and Section 30.16 “Generic Clock Controller” Figure 30-1 “General Clock Block Diagram”: added PLLA block; repositioned PLLACK signal; at bottom of diagram “PCKx” changed to “PCKx (to pads)” Table 30-3 “Register Mapping”: PMC_AUDIO_PLL0 reset value ‘0x0000_0000’ changed to ‘0x0000_00D0’
	Section 30. “Power Management Controller (PMC)” (cont”d) Section 30.22.11 “PMC Master Clock Register”: updated CSS field description Section 30.22.13 “PMC Programmable Clock Register”: added addresses 0xF0014044 and 0xF0014048; updated CSS field description Section 30.22.39 “PMC Audio PLL Control Register 0”: added fields DCO_FILTER (bits 29:28), DCO_GAIN (bits 27:24) and PLLFLT (bits 7:4) Section 30.22.40 “PMC Audio PLL Control Register 1”: updated DIV field description (cont’don next page)

Table 68-2. SAMA5D2 Datasheet Rev. 11267D Revision History (Continued)

Issue Date	Changes
12-May-16	Section 31. “Parallel Input/Output Controller (PIO)”
	Section 31.4.2 “External Interrupt Lines”: “are generally multiplexed” changed to “are multiplexed”
	Section 31.5 “Functional Description”: removed entire section “Peripheral Muxing Example”
	Table 31-4 “Register Mapping”:
	- added reset value for PIO_CFGR, PIO_ODSR, PIO_IMR, S_PIO_CFGR, S_PIO_ODSR and S_PIO_IMR
	- “PIO I/O Freeze Register” corrected to “PIO I/O Freeze Configuration Register”
	- defined offset range 0x400–0x4FC as reserved
	- reserved offset range 0x5E8–0x5F8 changed to 0x5E8–0x5FC
	- “Secure PIO I/O Freeze Register” corrected to “Secure PIO I/O Freeze Configuration Register”
	Removed duplicated or invalid addresses in Section 31.7.1 “PIO Mask Register”, Section 31.7.2 “PIO Configuration Register”, Section 31.7.3 “PIO Pin Data Status Register”, Section 31.7.4 “PIO Lock Status Register”, Section 31.7.5 “PIO Set Output Data Register”, and Section 31.7.6 “PIO Clear Output Data Register”
	Section 31.7.7 “PIO Output Data Status Register”: removed duplicated or invalid addresses; access “Read-only or Read/Write” corrected to “Read/Write”
	Removed duplicated or invalid addresses in Section 31.7.8 “PIO Interrupt Enable Register”, Section 31.7.9 “PIO Interrupt Disable Register”, Section 31.7.10 “PIO Interrupt Mask Register”, and Section 31.7.11 “PIO Interrupt Status Register”
	Section 31.7.12 “PIO I/O Freeze Configuration Register”: corrected title (was “PIO Freeze Configuration Register”); removed duplicated or invalid addresses; access “Read/Write” corrected to “Write-only”
	Removed duplicated or invalid addresses in Section 31.7.15 “Secure PIO Mask Register”, Section 31.7.16 “Secure PIO Configuration Register”, Section 31.7.17 “Secure PIO Pin Data Status Register”, Section 31.7.18 “Secure PIO Lock Status Register”, Section 31.7.19 “Secure PIO Set Output Data Register” and Section 31.7.20 “Secure PIO Clear Output Data Register”
	Section 31.7.21 “Secure PIO Output Data Status Register”: removed duplicated or invalid addresses; access “Read-only or Read/Write” corrected to “Read/Write”
	Removed duplicated or invalid addresses in Section 31.7.22 “Secure PIO Interrupt Enable Register”, Section 31.7.23 “Secure PIO Interrupt Disable Register”, Section 31.7.24 “Secure PIO Interrupt Mask Register”, and Section 31.7.25 “Secure PIO Interrupt Status Register”
	Section 31.7.29 “Secure PIO I/O Freeze Configuration Register”: corrected title (was “Secure PIO Freeze Configuration Register”); removed duplicated or invalid addresses; access “Read/Write” corrected to “Write-only”
	Section 31.7.30 “Secure PIO Slow Clock Divider Debouncing Register”: added sentence about register write protection
	Section 32. “External Memories”
	Table 32-1 “DDR/LPDDR I/O Lines Description”: updated DDR_VREF function description
	Section 33. “Multiport DDR-SDRAM Controller (MPDDRC)”
	Section 33.4.1 “Low-power DDR1-SDRAM Initialization”: in first paragraph, removed content about configuring register SFR_DDRCFG
	Section 33.6 “Software Interface/SDRAM Organization, Address Mapping”: modified description of Interleaved mode (“at each SDRAM end page” corrected to “at each DDRSDRAM end of page”)
	Harmonized register naming throughout Section 33.7 “AHB Multiport DDR-SDRAM Controller (MPDDRC) User Interface”
	Removed all MPDDRC DLL registers (offset range 0x100–0x158 now reserved)
	Section 33.7.3 “MPDDRC Configuration Register”: modified description of DECOD bit value ‘1’ (“at each SDRAM end page” corrected to “at each DDR-SDRAM end of page”)
	Section 33.7.12 “MPDDRC I/O Calibration Register”: updated RZQ values in RDIV field description

Table 68-2. SAMA5D2 Datasheet Rev. 11267D Revision History (Continued)

Issue Date	Changes
12-May-16	Section 34. “Static Memory Controller (SMC)” Section 34.17.3 “NFC Initialization”: instances of “rbn” changed to “Ready/Busy” Section 34.20.3 “NFC Status Register”: bit RB_EDGE3 (bit 27) replaced by RB_EDGE0 (bit 24); updated RB_RISE and RB_FALL bit descriptions Bit RB_EDGE3 (bit 27) replaced by RB_EDGE0 (bit 24) in Section 34.20.4 “NFC Interrupt Enable Register”, Section 34.20.5 “NFC Interrupt Disable Register” and Section 34.20.6 “NFC Interrupt Mask Register” Deleted invalid addresses in Section 34.20.30 “PMECC Error Location SIGMA0 Register” and Section 34.20.31 “PMECC Error Location SIGMAx Register” Section 34.20.32 “PMECC Error Location x Register”: register index “x=0..23” corrected to “x=0..31” Section 34.20.36 “Timings Register”: removed RBNSEL field
	Section 35. “DMA Controller (XDMAC)” Added XDMAC_CCx.CSIZE configuration to Table 35-2 “DMA Channels Definition (XDMAC0)” and Table 35-3 “DMA Channels Definition (XDMAC1)” Table 35-5 “Register Mapping”: - XDMAC_GCFG access Read-only corrected to Read/Write - XDMAC_GWAC access Read-only corrected to Read/Write Section 35.9.2 “XDMAC Global Configuration Register”: access Read-only corrected to Read/Write Section 35.9.3 “XDMAC Global Weighted Arbiter Configuration Register”: access Read-only corrected to Read/Write
	Section 36. “LCD Controller (LCDC)” Updated ”Section 36.2 “Embedded Characteristics” Updated Section 36.6.1.1 “Pixel Clock Period Configuration”
	Section 37. “Ethernet MAC (GMAC)” Table 37-1 “GMAC Connections in Different Modes”: added table Note on GTXCK Updated Section 37.5.3 “Interrupt Sources” Section 37.7.1.2 “Receive Buffer List” and Section 37.7.1.3 “Transmit Buffer List”: added note at end of sections on queue pointer intilaization Section 37.8.107 “GMAC Transmit Buffer Queue Base Address Register Priority Queue x” and Section 37.8.108 “GMAC Receive Buffer Queue Base Address Register Priority Queue x”: changed sentence on register initialization
	Section 39. “USB Host High Speed Port (UHPHS)” Section 39.2 “Embedded Characteristics”: “X Hosts (A and B) High Speed (EHCI)” corrected to “2 Hosts (A and B) High Speed (EHCI)” Table 39-2 “Register Mapping”: inserted reserved offset 0x0C Section 39.7.19 “EHCI: REG06 - AHB Error Status”: instances of “INSNREG[8:4]” changed to “INSNREG06[8:4]”
	Section 40. “Audio Class D Amplifier (CLASSD)” Section 40.2 “Embedded Characteristics”: DSP clock frequency “11.289 MHz” corrected to “11.2896 MHz” Section 40.5.2 “Power Management”: field name “NOVRLAP” corrected to “NOVRVAL” Figure 40-21 “Use Case 4B: Stereo Audio DAC With Passive Low Pass Filter and Single-ended Outputs”: changed title (was “Use Case 4B: Stereo Audio DAC With Passive Low Pass Filter and Differential Outputs”)
	Section 42. “Synchronous Serial Controller (SSC)” Figure 42-19 “Interrupt Block Diagram”: “RXSYNC” renamed to “RXSYN”; “TXSYNC” renamed to “TXSYN” Section 42.8.10 “Register Write Protection”: in first sentence, “AIC behavior” corrected to “SSC behavior”

Table 68-2. SAMA5D2 Datasheet Rev. 11267D Revision History (Continued)

Issue Date	Changes
12-May-16	Section 43. “Two-wire Interface (TWIHS)”
	Section 43.6.3.10 “SMBus Mode”: Deleted “A dedicated bus line, SMBALERT, allows a slave to get a master attention” from listed exceptions
	Section 43.6.5.6 “SMBus Mode”: Deleted “A dedicated bus line, SMBALERT, allows a slave to get a master attention” from listed exceptions
	Deleted note about debugger read access in Section 43.7.6 “TWIHS Status Register”, Section 43.7.13 “TWIHS Receive Holding Register” and Section 43.7.25 “TWIHS Write Protection Status Register”
	Section 44. “Flexible Serial Communication Controller (FLEXCOM)”
	Section 44.7.1.2 “Fractional Baud Rate in Asynchronous Mode”: in first paragraph, deleted sentence “This feature is only available when using USART Normal mode.”
	Figure 44-8 “Preamble Patterns, Default Polarity Assumed”: instances of “8 bit width” changed to “8-bit”
	Figure 44-11 “Asynchronous Start Detection”: added missing arrowheads
	Section 44.7.3.11 “Receiver Timeout”: removed redundant paragraphs on STTTO and RETTO; reworded two bullets
	Section 44.7.4 “ISO7816 Mode”: at end of second paragraph, “value 0x5 for protocol T = 1” changed to “value 0x6 for protocol T = 1”
	Section 44.7.4.2 “Protocol T = 0”: reworded content under “Receive NACK Inhibit”
	Section 44.7.7 “USART Comparison Function on Received Character”: modified information about the CMPMODE bit
	Table 44-18 “Register Mapping”: added TWI SleepWalking Matching Register (FLEX_TWI_SWMR)
	Section 44.10.41 “USART Write Protection Mode Register”: rephrased WPEN bit description
	Corrected order of all sections from Section 44.10.66 “TWI Interrupt Enable Register” to Section 44.10.76 “TWI SleepWalking Matching Register”
	Section 44.10.76 “TWI SleepWalking Matching Register”: added addresses
	Section 46. “Serial Peripheral Interface (SPI)”
	Figure 46-1 “Block Diagram”: added GCLK output from PMC to SPI
	Modified transmission condition description in Section 46.7.3 “Master Mode Operations”
	Section 46.7.4 “SPI Slave Mode”: added sentence about NSS rising between characters
	Section 46.7.5 “SPI Comparison Function on Received Character”: in seventh paragraph, added “if SleepWalking mode is disabled” to sentence “The comparison trigger event is...”
	Updated Section 46.7.8 “Register Write Protection”
	Section 46.8.2 “SPI Mode Register”: added bits BRSRCCLK (Bit Rate Source Clock) and LSBHALF (LSB Timing Selection); updated description of field DLYBCS
	Section 46.8.12 “SPI Chip Select Register”: updated description of fields CSNAAT, SCBR, DLYBS and DLYBCT
	Section 47. “Quad Serial Peripheral Interface (QSPI)”
	Section 47.2 “Embedded Characteristics”: added bullet “Interface to Serial Flash Memories Operating in Single Data Rate or Double Data Rate Modes”
	Section 47. “Quad Serial Peripheral Interface (QSPI)” (cont’d)
	NSS renamed to QCS in Figure 47-2 “QSPI Transfer Format (QSPI_SCR.CPHA = 0, 8 bits per transfer)” and Figure 47-3 “QSPI Transfer Format (QSPI_SCR.CPHA = 1, 8 bits per transfer)”
	Section 47.7.2 “QSPI Mode Register”: added note “This field is forced to LASTXFER when SMM is written to ‘1’ to CSMODE field description; modified equation in description of fields DLYBCT and DLYCS
	Section 47.7.5 “QSPI Status Register”: updated descriptions of bits CSR and INSTRE
	Section 47.7.9 “QSPI Serial Clock Register”: modified equation in description of fields SCBR and DLYBS

Table 68-2. SAMA5D2 Datasheet Rev. 11267D Revision History (Continued)

Issue Date	Changes
12-May-16	Section 48. “Secure Digital Multimedia Card Controller (SDMMC)” Added Section 48.3 “Embedded Features for SDMMC0 and SDMMC1” Figure 48-1 “Block Diagram”: added two notes Table 48-3 “Register Mapping”: updated SDMMC_APSR reset value (SDMMC0 different from SDMMC1) Section 48.13.18 “SDMMC Software Reset Register”: updated SWRSTCMD bit description Section 48.13.58 “SDMMC Calibration Control Register”: in CNTVAL field description, “$t_{STARTUP}$ = ...” corrected to “$t_{STARTUP}$ = 2 µs”
	Section 49. “Image Sensor Controller (ISC)” Added Section 49.4 “Product Dependencies” Table 49-18 “Register Mapping”: defined offset range 0x404–0x40C as reserved
	Section 50. “Controller Area Network (MCAN)” “GCLK3” changed to “GCLK” in Section 50.3 “Block Diagram” and Section 50.4.2 “Power Management” Added Table 50-2 “Peripheral IDs” Updated Section 50.5.1.3 “CAN FD Operation” Section 50.5.1.4 “Transmitter Delay Compensation”: modified title (was “Transceiver Delay Compensation”); revised content Section 50.5.1.5 “Restricted Operation Mode”: added ‘Note’ Section 50.5.3 “Timeout Counter”: “baud rate” changed to “bit rate” in ‘Note’ Section 50.5.4.1 “Acceptance Filtering”: “described in Section” corrected to “described in “Rx FIFO Overwrite Mode” Updated Figure 50-5 “Standard Message ID Filter Path”and Figure 50-6 “Extended Message ID Filter Path” Updated register names in Figure 50-7 “Rx FIFO Status” and Figure 50-8 “Rx FIFO Overflow Handling” Section 50.5.7.2 “Rx Buffer and FIFO Element”: “R1 Bit 21 FDF: Extended Data Length” renamed to “R1 Bit 21 FDF: FD Format” Section 50.5.7.4 “Tx Event FIFO Element”: “E1 Bit 21 FDF: Extended Data Length” renamed to “E1 Bit 21 FDF: FD Format” Table 50-14 “Register Mapping”: - deleted row “0x00–0x04 / Reserved” - “Fast Bit Timing and Prescaler Register” renamed to “Data Bit Timing and Prescaler Register” - “Bit Timing and Prescaler Register” renamed to “Nominal Bit Timing and Prescaler Register” Section 50.6.4 “MCAN Data Bit Timing and Prescaler Register”: changed name (was “MCAN Fast Bit Timing and Prescaler Register”); field FBRP replaced by field DBRP Section 50.6.7 “MCAN CC Control Register”: updated descriptions of fields FDOE, BRSE, PXHD and EFB; removed NISO bit Section 50.6.8 “MCAN Nominal Bit Timing and Prescaler Register”: “NBRP: Nominal Baud Rate Prescaler” changed to “NBRP: Nominal Bit Rate Prescaler” Section 50.6.9 “MCAN Timestamp Counter Configuration Register”: updated TSS field description Section 50.6.10 “MCAN Timestamp Counter Value Register”: updated TSC field description
	Section 50. “Controller Area Network (MCAN)” (cont’d) Section 50.6.20 “MCAN Global Filter Configuration”: added details on register description; updated ANFE and ANFS field descriptions. Added details on register description in Section 50.6.21 “MCAN Standard ID Filter Configuration” and Section 50.6.22 “MCAN Extended ID Filter Configuration” Section 50.6.24 “MCAN High Priority Message Status”: updated description of MSI field value ‘1’

Table 68-2. SAMA5D2 Datasheet Rev. 11267D Revision History (Continued)

Issue Date	Changes
12-May-16	Section 51. “Timer Counter (TC)” Replaced TIOA, TIOB, TCLK with TIOAx, TIOBx, TCLKx Table 51-1 “Timer Counter Clock Assignment”: updated definitions Section 51.6.3 “Clock Selection”: updated bullet “Internal clock signals”, updated notes 1 and 2 Section 51.6.9 “Transfer with DMAC in Capture Mode”: updated title (added “in Capture Mode”) Updated Figure 51-5 “Example of Transfer with DMAC in Capture Mode” Section 51.6.16.4 “Position and Rotation Measurement”: updated text in first paragraph Added Section 51.6.16.6 “Detecting a Missing Index Pulse” Updated TCCLKS field description in Section 51.7.2 “TC Channel Mode Register: Capture Mode” and Section 51.7.3 “TC Channel Mode Register: Waveform Mode”
	Section 53. “Pulse Width Modulation Controller (PWM)” Throughout, “PWMTRG” and “EXTTRG” renamed to “PWMEXTRG” Table 53-2 “I/O Lines”: “PWMEXTRG0” and “PWMEXTRG1” renumbered to “PWMEXTRG1” and “PWMEXTRG2” Updated Section “Recoverable Fault” Updated Figure 53-1 “Pulse Width Modulation Controller Block Diagram” and added note below figure Updated Figure 53-16 “Fault Protection”
	Section 54. “Secure Fuse Controller (SFC)” Table 54-1 “Register Mapping”: removed reset value from SFC_IER and SFC_IDR (both registers are write-only)
	Section 55. “Integrity Check Monitor (ICM)” Table 55-8 “Register Mapping”: ICM_SR access “Write-only” corrected to “Read-only”
	Section 57. “Advanced Encryption Standard (AES)” Table 57-5 “Register Mapping”: AES_ALPHAR[0..3] access “Write” corrected to “Write-only” Section 57.5.20 “AES Alpha Word Register x”: access “Write” corrected to “Write-only”
	Section 59. “Triple Data Encryption Standard (TDES)” Section 59.4.1 “Operating Modes”: deleted sentence “The OFB and CFB modes of operation are only available if 2-key mode is selected (KEYMOD = 1 in TDES_MR).” Section 59.4.3 “Last Output Data Mode”: deleted sentence “No more Output Data Register reads are necessary between consecutive encryptions/decryptions (see Section 59.4.3 “Last Output Data Mode”).” Section 59.5.2 “TDES Mode Register”: in OPMOD field description, deleted sentence “The OFB and CFB modes of operation are only available if 2-key mode is selected (KEYMOD = 1).”
	Section 61. “Security Module” Updated Figure 61-2 “Security Module Internal Memory Map”
	Section 61. “Analog-to-Digital Converter (ADC)” Section 61.1 “Description”: - deleted sentence “A digital error correction circuit based on the multi-bit redundant signed digit (RSD) algorithm is implemented to reduce INL and DNL errors.” - deleted sentence “Finally, the user can configure ADC timings, such as startup time and tracking time.”

Table 68-2. SAMA5D2 Datasheet Rev. 11267D Revision History (Continued)

Issue Date	Changes
12-May-16	Section 61. “Analog-to-Digital Converter (ADC)” (cont’d) Updated Section 61.2 “Embedded Characteristics” Updated Figure 61-1 “Analog-to-Digital Converter Block Diagram” Revised Section 61.5 “Product Dependencies” Revised Section 61.6.1 “Analog-to-Digital Conversion” Updated Section 61.6.3 “ADC Reference Voltage” and Section 61.6.4 “Conversion Resolution” Updated Section 61.6.7 “Conversion Triggers” Section 61.6.9 “Comparison Window”: in fourth paragraph, instance of “ADC_SR” corrected to “ADC_ISR” Section 61.6.10 “Differential and Single-ended Input Modes”: changed title (was “Differential Inputs”) and revised content Updated Section 61.6.11 “ADC Timings”, Section 61.6.12 “Last Channel Specific Measurement Trigger”, Section 61.6.13 “Enhanced Resolution Mode and Digital Averaging Function” and Section 61.6.14 “Automatic Error Correction” Instances of GND renamed to GNDANA in Figure 61-15 “Touchscreen Switches Implementation”, Figure 61-18 “Touchscreen Switches Implementation” and Figure 61-20 “Touchscreen Pen Detect” Updated Section 61.6.16 “Asynchronous and Partial Wakeup (SleepWalking)” Section 61.6.17.1 “Classic ADC Channels Only (Touchscreen Disabled)”: changed title (was “Classical ADC Channels Only”) Section 61.6.19 “Register Write Protection”: updated list of protectable registers Table 61-8 “Register Mapping”: - defined 0x48 as reserved - added row 0x4C / Channel Offset Register / ADC_COR - added offset 0x7C for ADC_CDR11 - defined offset range 0x80–0x90 as reserved - added row 0x94 / Analog Control Register / ADC_ACR - defined offset range 0xC4–0xD0 as reserved - added row 0xD4 / Correction Values Register / ADC_CVR - added row 0xD8 / Channel Error Correction Register / ADC_CECR - added row 0xDC / Touchscreen Correction Values Register / ADC_TSCVR - defined offset 0xE0 as reserved Section 61.7.2 “ADC Mode Register”: updated TRACKIM field description Added LCCHG (Last Channel Change) bit in Section 61.7.9 “ADC Interrupt Enable Register”, Section 61.7.10 “ADC Interrupt Disable Register”, Section 61.7.11 “ADC Interrupt Mask Register” and Section 61.7.12 “ADC Interrupt Status Register” Section 61.7.13 “ADC Last Channel Trigger Mode Register”: updated CMPMOD field description Section 61.7.16 “ADC Extended Mode Register”: updated CMPMODE field description; added descriptions for fields OSR ASTE Section 61.7.18 “ADC Channel Offset Register”: added address; removed bits OFF11:OFF0 from bitmap; modified DIFFx field description Section 61.7.20 “ADC Analog Control Register”: added address; added IBCTL field Section 61.7.25 “ADC Trigger Register”: added sentence about write protection Removed Section “Correction Select Register” Added sentence about write protection in Section 61.7.26 “ADC Correction Values Register” and Section 61.7.27 “ADC Channel Error Correction Register” Added Section 61.7.28 “ADC Touchscreen Correction Values Register”

Table 68-2. SAMA5D2 Datasheet Rev. 11267D Revision History (Continued)

Issue Date	Changes
12-May-16	Section 62. “Electrical Characteristics” “ADVREFP” corrected to “ADVREF” Section 62.2 “DC Characteristics”: in first sentence, “T_A = -40°C to +85°C” changed to “T_A = -40°C to +105°C” Added Table 62-2 “Recommended Thermal Operating Conditions” Updated Section 62.4 “Active Mode” Table 62-8 “Low-power Mode Configuration Summary”: updated values for ‘Consumption’ and ‘Wakeup Time’ Updated Section 62.5.6 “Low-power Consumption Versus Modes” Table 62-9 “Typical Power Consumption in Idle Mode: AMP2”: updated consumption values Table 62-10 “VDDCORE Power Consumption in Ultra Low-power Mode: AMP2”: updated consumption values; updated wakeup time for ULP1 Fast Wakeup mode Table 62-11 “Typical Power Consumption for Backup Mode”: updated consumption values Updated Table 62-12 “Processor Clock Waveform Parameters” and Table 62-13 “Master Clock Waveform Parameters” Updated Section 62.7.1 “Main Oscillator Characteristics” Table 62-17 “12 MHz RC Oscillator Characteristics”: updated startup time values Updated Section 62.7.3 “32.768 kHz Crystal Oscillator Characteristics” Updated Table 62-23 “Audio PLL Characteristics” Section 62.10 “ADC Characteristics”: deleted sentence “The VREFN pin must be connected to ground.” Table 62-25 “Power Supply Characteristics”: updated Analog Current Consumption value for Fast Wakeup mode Table 62-26 “ADVREF Electrical Characteristics”: “VREFP” corrected to “ADVREF”; updated Current value Section 62.10.4.1 “Differential Mode (12-bit mode)” and Section 62.10.4.2 “Single-ended Mode (12-bit mode)”: in equation, “V_{VREFP}” corrected to “V_{ADVREF}” Section 62.10.4.4 “Gain and Offset Errors”: “V_{VREFP}” corrected to “V_{ADVREF}” Table 62-27 “ADC Timing Characteristics”: updated footnote Added Table 62-32 “ADC Analog Input Characteristics” Table 62-37 “VDDCORE Power-On Reset Characteristics”: updated Hysteresis Voltage values Section 62.14.1 “Maximum SPI Frequency”: updated values in “Master Read Mode” and “Slave Write Mode” Revised Section 62.18 “MPDDRC Timings” Corrected CKI values in Figure 62-33 “SSC Transmitter, TK and TF in Input”, Figure 62-35 “SSC Receiver, RK in Input and RF in Output”, Figure 62-36 “SSC Receiver, RK and RF in Output” and Figure 62-38 “Minimum and Maximum Access Time of Output Signals”
	Section 64. “Schematic Checklist” Figure 64-1 “1.2V, 1.35V/1.5V, 2V, 2.5V, 3.3V Power Supplies Schematics(1)”: GNDHSIC changed to GNDUTMIC Table 64-1 “Power Supply Connections”: updated GNDUTMIC row; removed GNDHSIC row; in second footnote, “microcontroller” changed to “microprocessor” Table 64-2 “Clock, Oscillator and PLL Connections”: “(internal 32-kHz RC oscillator) changed to “(internal 64 kHz RC oscillator)” Section 64.5.1 “How to Define the Oscillator Load Capacitance”: instances of “32-KHz Oscillator” changed to “32.768 kHz Oscillator” Added Section 64.14.6 “QSPI Pull-up Resistors”
	Updated Section 66. “Ordering Information”
	Section 67. “Errata” Updated content (errata now collected in Section 67.1 “Errata - SAMA5D2 MRL-B Parts” and Section 67.2 “Errata - SAMA5D2 MRL-A Parts”)

Table 68-3. SAMA5D2 Datasheet Rev. 11267C Revision History

Issue Date	Changes
8-Jan-16	Changed datasheet status from ‘Preliminary’ to ‘Complete’.
	Added “Introduction” and transferred Description to Section 1.
	Section 2. “Configuration Summary” Added device compatibility information
	Section 4. “Signal Description” Table 4-1 “Signal Description List”: modified rows PIOBU 0-7 and DDR_RESETN
	Section 6. “Power Considerations” Added Section 6.4.1 “VDDBU Power Architecture” Updated Section 6.2 “Powerup Considerations” and Section 6.3 “Powerdown Considerations”
	Section 7. “Memories” Updated Section 7.1.2 “Internal ROM”
	Section 10. “Peripherals” Updated Table 10-1 “Peripheral identifiers” and Section 10.4 “Peripheral Clock Types”
	Section 16. “AXI Matrix (AXIMX)” Table 16-1 “Register Mapping”: removed 0x00000000 reset value from all rows
	Section 17. “Matrix (H64MX/H32MX)” Section 17.2 “Embedded Characteristics”: removed "Master number forwarding to slaves" characteristic Updated Table 17-1 “List of H64MX Masters”, Table 17-2 “List of H64MX Slaves”, Table 17-4 “List of H32MX Masters”, Table 17-5 “List of H32MX Slaves” Table 17-3 “Master to Slave Access on H64MX”: updated ‘SDMMC0-SDMMC1’ row Table 17-6 “Master to Slave Access on H32MX”: updated ‘Slave 5’ rows Section 17.12.2 “Security of APB Slaves”: added introduction and bulleted list introduced by “As a general rule” Added Section 17.12.3 “Security Types of AHB Master Peripherals” and Section 17.12.4 “Security Types of AHB Slave Peripherals” Section 17-9 “Peripheral Identifiers”: corrected some security type names Section 17.13 “AHB Matrix (MATRIX) User Interface”: added introduction and modified reset value of Updated Security Areas Split Slave x Registers in Table 17-10 “Register Mapping”
	Section 24. “Watchdog Timer (WDT)” Replaced “Idle mode” with “Sleep mode (Idle mode)” in Section 24.1 “Description” and with “Sleep mode” in Section 24.4 “Functional Description”
	Section 22. “Reset Controller (RSTC)” Renamed 'proc_nreset' to 'Processor Reset', 'periph_nreset' to 'Peripheral Reset', 'backup_neset' to 'Backup Reset', 'rstc_irq' to 'Reset Controller Interrupt', 'wd_fault' to 'Watchdog Fault', ‘user_reset’ to User Reset. Updated text and figures to show that Processor Reset and Peripheral Reset signals are merged.
	Section 23. “Shutdown Controller (SHDWC)” Updated Figure 23-1 “Shutdown Controller Block Diagram” and Table 23-1 “I/O Lines Description” Section 23.7.3 “Shutdown Status Register”: corrected register table (added WKUPIS9) Section 23.7.4 “Shutdown Wakeup Inputs Register”: corrected register table (added WKUPT9 and WKUPEN9)

Table 68-3. SAMA5D2 Datasheet Rev. 11267C Revision History (Continued)

Issue Date	Changes
8-Jan-16	Section 29. “Real-time Clock (RTC)” Removed RTC Milliseconds Register (RTC_MSR) and all related information in Section 29.1 “Description”, Section 29.2 “Embedded Characteristics”, Section 29.5 “Functional Description” and Section 29.6 “Real-time Clock (RTC) User Interface”. Table 29-1 “Register Mapping”: modified RTC_CALR reset value Section 29.6.1 “RTC Control Register”: updated CALEVSEL field description Updated Section 29.6.22 “RTC TimeStamp Source Register”
	Section 29. “Clock Generator” Section 29.2 “Embedded Characteristics”: replaced "400 to 1000 MHz programmable PLL" with "600 to 1200 MHz programmable PLL" and replaced “HCLOCK” with “HCLOCK_LS/HS” and “PCLOCK” with “PCLOCK_LS/HS” Section 29.4 “Slow Clock”: removed “This allows the slow clock to be valid in a short time (about 100 µs)” Section 29.8 “Audio PLL”: updated all equations and added “in the 700 MHz range” after “The PLL core operates at 700 MHz (AUDIOCORECLOCK)” Updated Figure 29-3. Main Clock Block Diagram and Figure 29-4. Main Clock Source Selection
	Section 30. “Power Management Controller (PMC)” Updated Section 30.6 “Matrix Clock Controller” Updated Section 30-1 “General Clock Block Diagram” Section 30.19 “Programming Sequence”, sub-section “Selecting Master Clock and Processor Clock”: updated sequence following "If a new value for CSS field corresponds to PLL Clock" Section 30.22.11 “PMC Master Clock Register”: updated H32MXDIV field description
	Section 33. “Multi-port DDR-SDRAM Controller (MPDDRC)” Section 33-2 “Single Write Access, Row Closed, DDR-SDRAM Devices” to Section 33-8 “SINGLE Write Access Followed by a Read Access, DDR2-SDRAM Devices”: replaced “D[15:0]” with “DATA” Updated Section 33.7.9 “MPDDRC Low-power DDR2 Low-power DDR3 Low-power Register” Section 33.7.10 “MPDDRC Low-power DDR2 Low-power DDR3 and DDR3 Calibration and MR4 Register”: updated MR4_READ field description
	Section 34. “Static Memory Controller (SMC)” Removed NFCCMD field and modified Section 34.17.2.1 “Building NFC Address Command Example” and Section 34.17.2.2 “NFC Address Command” accordingly Table 34-20 “Register Mapping”: corrected offset values of PMECC Error Location 31 Register and of subsequent reserved range; removed reset value from HSMC_CTRL (register is write-only)
	Section 42. “DMA Controller (XDMAC)” Section 42.5.4.1 “Single Block With Single Microblock Transfer”: added text on memory-to-memory transfer Section 42.8 “XDMAC Software Requirements”: added bullet on memory-to-memory transfer Table 42-5 “Register Mapping”: corrected access of XDMAC_GTYPE, XDMAC_GWAC, XDMAC_CIM Section 42.9.6 “XDMAC Global Interrupt Mask Register”: corrected access to Read-only Section 42.9.28 “XDMAC Channel x [x = 0..15] Configuration Register”: corrected INITD and PERID field descriptions
	Section 36. “LCD Controller (LCDC)” Modified width of fields in Section 36.7.2 “LCD Controller Configuration Register 1” and Section 36.7.3 “LCD Controller Configuration Register 2”
	Section 40. “Audio Class D Amplifier (CLASSD)” Replaced ‘audio clock’ with ‘generic clock’ and ‘ACLK’ with ‘GCLK’ throughout the section

Table 68-3. SAMA5D2 Datasheet Rev. 11267C Revision History (Continued)

Issue Date	Changes
8-Jan-16	Section 41. “Inter-IC Sound Controller (I2SC)” Section 41.6.3 “Master, Controller and Slave Modes”: removed text fragment: ‘in order to avoid unwanted glitches on the I2SWS and I2SCK pins.’ Section 41.8.2 “Inter-IC Sound Controller Mode Register”: removed text fragment: ‘in order to avoid unexpected behavior on the I2SWS, I2SCK and I2SDO outputs.’ and added note (2) below IMCKDIV field description.
	Section 44. “Flexible Serial Communication Controller (FLEXCOM)” Restored all references to ISO7816 specification Updated Figure 44-3 “Fractional Baud Rate Generator” Added Figure 44-27 “RTS line software control when FLEX_US_MR.USART_MODE = 2” Section 44.10.6 “USART Mode Register”: updated USART_MODE field description (SPI_MASTER item) Section 44.10.44 “SPI Mode Register”: added LBHPC bit
	Section 55. “Universal Asynchronous Receiver Transmitter (UART)” Section 55.6.9 “UART Baud Rate Generator Register”: in CD field description, corrected equation after “If BRSRCCK = 1”
	Section 59. “Quad SPI Interface (QSPI)” Section 59.7.5 “QSPI Status Register”: updated RDRF, TDRE, TXEMPTY, and OVRES field descriptions
	Section 48. “Secure Digital Multimedia Card Controller (SDMMC)” Section 48.12.41 “SDMMC Preset Value Register”: updated CLKGSEL field description
	Section 49. “Image Sensor Controller (ISC)” Section 49.1 “Description”: removed "serial csi-2 based CMOS/CCD sensor" (not supported).
	Section 50. “Controller Area Network (MCAN)” Changed MCAN interrupt line names to MCAN_INT0 and MCAN_INT1 thoughout the section Section 50.6.7 “MCAN CC Control Register”: added bit NISO
	Section 51. “Timer Counter (TC)” Reformatted and renamed Table 51-2 “Channel Signal Description” Section 51.6.3 “Clock Selection”: updated notes (1) and (2)
	Section 52. “Pulse Density Modulation Interface Controller (PDMIC)” Replaced all instances of “PCK” with “GCLK” Section 52.2 “Embedded Characteristics”: removed ‘Multiplexed PDM Input Support’ characteristic Updated Section 52.5.2 “Power Management” and Section 52.6.2.1 “Description” Section 52.6.2.6 “Gain and Offset Compensation”: updated *dgain* bullet Section 52.7.3 “PDMIC Converted Data Register”: updated DATA field description Section 52.7.8 “PDMIC DSP Configuration Register 0”: updated OSR field description
	Section 61. “Security Module” Section 61.5.5 “SECUMOD Status Clear Register”: removed MCKM field description Section 61.5.18 “SECUMOD Wake Up Register”: removed TPML field description
	Section 77. “Analog-to-Digital Converter (ADC)” Section 77.7.2 “ADC Mode Register”: updated TRACKTIM and TRANSFER field descriptions.

Table 68-3. SAMA5D2 Datasheet Rev. 11267C Revision History (Continued)

Issue Date	Changes
8-Jan-16	Section 62. “Electrical Characteristics” Updated tables from Table 62-3 “DC Characteristics” to Table 62-35 “Analog Comparator Characteristics” Updated Figure 62-3 “Main Oscillator Schematics” Corrected Gain Error formula under Figure 62-6 “Gain and Offset Errors in Single-ended Mode” Removed Figure 63-4 “Single-ended Mode ADC” and Figure 63-5 “Differential Mode ADC” Updated wake-up pin numbers in Section 62.5.1 “Backup Mode” and Section 62.5.3.2 “ULP1 Mode” Updated Section 62.5.4 “Idle Mode” and Section 62.23 “SDMMC Timings” Section 62.14.3 “Timing Extraction”: added introduction and Figure 62-12 “MISO Capture in Master Mode”
	Section 64. “Schematic Checklist” Removed Table 65-12. “EBI Pins and NAND Flash Device Connections” and Table 65-13. “DDR2 I/O Lines Usage vs Operating Modes”
	Reorganized Section 66. “Ordering Information”
	Updated Section 67. “Errata”

Table 68-4. SAMA5D2 Datasheet Rev. 11267B Revision History

Issue Date	Changes
13-Nov-15	"Features" Updated Security features
	Section 3. "Block Diagram" Updated Figure 3-1 "SAMA5D2 Series Block Diagram".
	Section 5. "Package and Pinout" Updated Table 5-2 "Pin Description (SAMA5D21, SAMA5D22, SAMA5D24, SAMA5D26, SAMA5D27, SAMA5D28A)" Removed Section 4.2 "Input/Output Description" and Section 4-3 "SAMA5D2 I/O Type Description"
	Section 6. "Power Considerations" Updated Table 6-1 "SAMA5D2 Power Supplies" Updated Figure 6-1 "Recommended Powerup Sequence", Figure 6-2 "Recommended Powerdown Sequence", Figure 6-3 "Recommended Backup Mode Entry", Figure 6-4 "Recommended Power Supply Sequencing at Wakeup"
	Section 8. "Event System" Updated Table 8-1 "Real-time Event Mapping List"
	Section 15. "Standard Boot Strategies" Replaced all instances of "GPBR" with "BUREG".
	Section 20. "Special Function Registers (SFR)" Updated Section 20.3.15 "I2S Register"
	Section 20. "Advanced Interrupt Controller (AIC)" Removed Sections "Interrupt Vectoring" and "Fast Interrupt Vectoring" Updated Section 20.8.3.3 "Interrupt Handlers"and Section 20.8.4.3 "Fast Interrupt Handlers"
	Section 29. "Power Management Controller (PMC)" Replaced "generated clock" with "generic clock", and "GCK" with "GCLK" Updated Section 29.22.8 "PMC Clock Generator Main Oscillator Register"
	Section 37. "Parallel Input/Output Controller (PIO)" Removed all references to programmable I/O delay
	Added Section 32. "External Memories"
	Section 33. "Multi-port DDR-SDRAM Controller (MPDDRC)" Section 33.4.3 "Low-power DDR2-SDRAM Initialization": added Step 14., Step 15. and Step 21. Section 33.4.5 "Low-power DDR3-SDRAM Initialization": added Step 14., Step 15. and Step 21. Section 33.7.8 "MPDDRC Memory Device Register": updated DBW field description; corrected location of fields RL3 and WL
	Section 37. "Ethernet MAC (GMAC)" Updated Section 37.1 "Description" Section 37.5.2 "Power Management": deleted reference to PMC_PCER Section 37.5.3 "Interrupt Sources": deleted reference to 'Advanced Interrupt Controller'. Replaced by 'Interrupt Controller'. Section 37.6.14 "IEEE 1588 Support": deleted reference to GMAC_TSSx. Removed reference to 'output pins' in 2nd paragraph. Section 37.6.15 "Time Stamp Unit": added information on GTSUCOMP signal in last paragraph

Table 68-4. SAMA5D2 Datasheet Rev. 11267B Revision History (Continued)

Issue Date	Changes
13-Nov-15	Section 39. “Audio Class D Amplifier (CLASSD)” Updated Figure 39-1. CLASSD Block Diagram
	Section 41. “Inter-IC Sound Controller (I2SC)” Replaced all instances of “PCKx” with “GCLK” Removed all references to Time Division Multiplexed (TDM) format (not supported) Section 41.1 “Description”: replaced “The I2SC can use either a single DMA Controller channel for both audio channels or one DMA Controller channel per audio channel.” with “The I2SC uses a single DMA Controller channel for both audio channels.”, and updated Section 41.2 “Embedded Characteristics” and Section 41.6.8 “DMA Controller Operation” accordingly Section 41.8.2 “Inter-IC Sound Controller Mode Register”: removed fields RXDMA and TXDMA
	Section 44. “Flexible Serial Communication Controller (FLEXCOM)” Added SPI mode in UART/USART Replaced all instances of ‘PCK’ with ‘GCLK’ Replaced all instances of ‘DMAC/PDC’ with ‘DMAC’ Removed SleepWalking characteristic from UART/USART mode Removed all references to ISO7816 specification Section 44.10.6 “USART Mode Register”updated USCLKS field description Section 44.10.44 “SPI Mode Register”: updated BRSRCCLK and DLYBCS field descriptions Section 44.10.54 “SPI Chip Select Register”: updated CSNAAT, SCBR, DLYBS and DLYBCT field descriptions Section 44.10.64 “TWI Clock Waveform Generator Register”: updated BRSRCCLK and CKSRC field descriptions Updated Figure 44-1 “FLEXCOM Block Diagram” and Figure 44-67 “Master Mode Block Diagram”
	Section 42. “Two-wire Interface (TWIHS)” Replaced all instances of “PMC_PCK” with “GCLK”
	Section 55. “Universal Asynchronous Receiver Transmitter (UART)” Replaced "Processor-Independent Source Clock" with "Processor-Independent Generic Source Clock" and "PCK" with "GCLK"
	Section 48. “Secure Digital Multimedia Card Controller (SDMMC)” Updated revision of supported e.MMC specification (from V4.41 to V4.51)
	Section 51. “Pulse Density Modulation Interface Controller (PDMIC)” Removed all references to PDC Removed Section 1.6.4 “Buffer Structure”
	Section 54. “Secure Fuse Controller (SFC)” Removed all references to lock fuse (not supported) Section 54.4.5.3 “Fuse Masking”: corrected data register names Section 54.5.2 “SFC Mode Register”: updated MSK field description Table 54-1 “Register Mapping”: modified SFC_IER and SFC_IDR access type from “Read/Write” to “Write-only”
	Section 57. “Advanced Encryption Standard (AES)” Updated Figure 57-12 “Generation of an ESP IPSec Frame without ESN” and Figure 57-13 “Generation of an ESP IPSec Frame with ESN”
	Added Section 61. “Security Module”

Table 68-4. SAMA5D2 Datasheet Rev. 11267B Revision History (Continued)

Issue Date	Changes
13-Nov-15	Section 61. “Analog-to-Digital Converter (ADC)” Updated enhanced resolution value from 12 bits to 14 bits Renamed “Hold time” to “Transfer time” Replaced all instances of “PMC PCK” with “GCLK” Added Section 61.6.6 “Conversion Results Format”, Section 61.7.13 “ADC Last Channel Trigger Mode Register”, Section 61.7.14 “ADC Last Channel Compare Window Register” Section 61.2 “Embedded Characteristics”: corrected conversion rate Section 61.6.9 “Comparison Window”: added paragraph about bit SIGNMODE Section 61.6.14 “Automatic Error Correction”: replaced “GAIN_ERROR_SIZE-1” with appropriate value; replaced “Gs-1” with “Gs” in formulas Section 61.6.14 “Automatic Error Correction”, Section 62.7.27 “Correction Values Register”: replaced “GAIN_ERROR_SIZE-1” and “OFFSET_ERROR_SIZE-1” with appropriate values Section 61.7.2 “ADC Mode Register”: updated TRGSEL and TRACKTIM field descriptions Updated Section 61.7.8 “ADC Last Converted Data Register” Section 61.7.16 “ADC Extended Mode Register”: added bit SIGNMODE Updated Section 61.7.18 “ADC Channel Offset Register” Updated Section 61-1 “Analog-to-Digital Converter Block Diagram” and Section 61-7 “Analog Full Scale Ranges in Single-Ended/Differential Applications” Updated Table 62-5 “Oversampling Digital Output Range Values”
	Section 62. “Electrical Characteristics” Added: - Section 62.11 “Analog Comparator Characteristics” - Section 62.14.1 “Maximum SPI Frequency” - Section 62.16.1 “Maximum QSPI Frequency” - Table 62-4 “I/O Switching Frequency” - Table 62-5 “QSPI I/O Switching Frequency” - Table 62-22 “UTMI PLL Characteristics” - Table 62-23 “Audio PLL Characteristics” Updated: - Table 62-1 “Absolute Maximum Ratings*” - Table 62-3 “DC Characteristics” - Table 63-8 “Typical Peripheral Power Consumption by Peripheral in Active Mode” to Table 62-11 “Typical Power Consumption for Backup Mode” - Table 62-14 “8 to 24 MHz Crystal Oscillator Characteristics” - Table 62-17 “12 MHz RC Oscillator Characteristics” to Table 62-22 “UTMI PLL Characteristics” - Table 62-36 “VDDBU Power-On Reset Characteristics” to Table 62-38 “VDDANA Power-On Reset Characteristics” Reworked Section 62.9 “USB HS Characteristics”
	Section 64. “Schematic Checklist” Updated: - Section 64.14.3 “DDR Layout and Design Considerations” - Figure 64-1 “1.2V, 1.35V/1.5V, 2V, 2.5V, 3.3V Power Supplies Schematics(1)” - Table 64-1 “Power Supply Connections”

Table 68-5. SAMA5D2 Datasheet Rev. 11267A Revision History

Issue Date	Changes
10-Sep-15	Preliminary Datasheet - First issue
25-Feb-15	Advance Information Datasheet.

Table of Contents

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**